



JUNOS™
Internet Software
Configuration Guide

VPNs

Release 5.6

Juniper Networks, Inc.

1194 North Mathilda Avenue
Sunnyvale, CA 94089

USA

408-745-2000

www.juniper.net

Part Number: 530-008942-01, Revision 1

Juniper
Networks

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986–1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by The Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, The Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. Gated is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of Gated has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

This product includes software developed by Maker Communications, Inc., Copyright © 1996, 1997, Maker Communications, Inc.

Juniper Networks is registered in the U.S. Patent and Trademark Office and in other countries as a trademark of Juniper Networks, Inc. Broadband Cable Processor, ERX, ESP, G10, Internet Processor, JUNOS, JUNOScript, M5, M10, M20, M40, M40e, M160, MRX, M-series, NMC-RX, SDX, ServiceGuard, T320, T640, T-series, UMC, and Unison are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners. All specifications are subject to change without notice.

JUNOS Internet Software Configuration Guide: VPNs, Release 5.6
Copyright © 2002, Juniper Networks, Inc.
All rights reserved. Printed in USA.

Writer: Albert Statti
Editor: Stella Hackell
Covers and template design: Edmonds Design

Revision History
27 December 2002—First edition.

The information in this document is current as of the date listed in the revision history.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer or otherwise revise this publication without notice.

Products made or sold by Juniper Networks (including the M5, M10, M20, M40, M40e, and M160 routers, T320 router, T640 routing node, and the JUNOS software) or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,333,650, 6,359,479, and 6,406,312.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. The JUNOS software has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

SOFTWARE LICENSE

The terms and conditions for using this software are described in the software license contained in the acknowledgment to your purchase order or, to the extent applicable, to any reseller agreement or end-user purchase agreement executed between you and Juniper Networks. By using this software, you indicate that you understand and agree to be bound by those terms and conditions.

Generally speaking, the software license restricts the manner in which you are permitted to use the software and may contain prohibitions against certain uses. The software license may state conditions under which the license is automatically terminated. You should consult the license for further details.

For complete product documentation, please see the Juniper Networks Web site at www.juniper.net/techpubs.

Table of Contents

About This Manual

Objectives	xix
Audience	xx
Document Organization	xx
Part Organization	xxi
Using the Indexes	xxii
Documentation Conventions	xxiii
General Conventions	xxiii
Conventions for Software Commands and Statements	xxiii
List of Technical Publications	xxv
Documentation Feedback	xxv
How to Request Support	xxvi

Part 1

VPN Overview

Chapter 1

VPN Overview	3
VPN Terminology	3
Differences between Layer 2 and Layer 3 VPNs	4
VPN Graceful Restart	4

Part 2

Layer 2 VPNs

Chapter 2

Layer 2 VPN Overview	9
Layer 2 VPN Overview	9
Layer 2 VPN Standards	10

Chapter 3

Layer 2 VPN Configuration Guidelines 11

Configure MPLS LSPs between the PE Routers	12
Configure MPLS LSPs Using LDP	12
Configure MPLS LSPs Using RSVP	14
Configure an IGP on the PE Routers	15
Configure an IBGP Session between PE Routers	16
Configure Routing Instances for Layer 2 VPNs on the PE Routers	16
Configure the Description	17
Configure the Instance Type	17
Configure Interfaces for Layer 2 VPN Routing	18
Configure CCC Encapsulation on Interfaces	18
Configure TCC Encapsulation on Interfaces	19
Configure Layer 2 VPN Policing on Interfaces	20
Configure the Route Distinguisher	21
Disable the Control Word for Layer 2 VPNs	22
Configure Policy for the Layer 2 VPN Routing Table	22
Configure the Connections to the Local Site	23
Configure the Site	23
Configure the Remote Site ID	24
Configure the Encapsulation Type	24
Trace Layer 2 VPN Traffic and Operations	25
Disable Normal TTL Decrementing for VPNs	26
Configure Packet Forwarding for VPNs	26
Configure VPN Graceful Restart	27

Chapter 4

Layer 2 VPN Configuration Example 29

Simple Full-Mesh Layer 2 VPN Overview	29
Enable an IGP on the PE Routers	30
Configure MPLS LSP Tunnels between the PE Routers	31
Configure IBGP on the PE Routers	32
Configure Routing Instances for Layer 2 VPNs on the PE Routers	34
Configure CCC Encapsulation on the Interfaces	36
Configure VPN Policy on the PE Routers	37
Layer 2 VPN Configuration Summarized by Router	39
Summary for Router A (PE Router for Sunnyvale)	39
Summary for Router B (PE Router for Austin)	41
Summary for Router C (PE Router for Portland)	43

Chapter 5

Summary of Layer 2 VPN Configuration Statements 47

control-word	47
description	47
encapsulation	48
encapsulation (physical interface)	48
encapsulation (logical interface)	49
encapsulation (Layer 2 VPN)	50

instance-type	51
interface	51
l2vpn	51
no-control-word	51
proxy	52
remote	52
remote-site-id	52
route-distinguisher	53
route-distinguisher-id	53
site	54
site-identifier	54
traceoptions	55
vrf-export	56
vrf-import	56

Part 3 Layer 3 VPNs

Chapter 6 Layer 3 VPN Overview 59

Layer 3 VPN Overview	59
Layer 3 VPN Standards	60
Layer 3 VPN Attributes	60
VPN-IPv4 Addresses and Route Distinguishers	61
IPv6 Layer 3 VPNs	64
VPN Routing and Forwarding Tables	65
Route Distribution within a Layer 3 VPN	68
Distribution of Routes from CE to PE Routers	69
Distribution of Routes between PE Routers	70
Distribution of Routes from PE to CE Routers	71
Forwarding across the Provider's Core Network	72
Routing Instances for VPNs	73
Multicast over Layer 3 VPNs	73
Multicast over Layer 3 VPNs Overview	74
Sending PIM Hello Messages to the PE Routers	75
Sending PIM Join Messages to the PE Routers	76
Receiving the Multicast Transmission	77

Chapter 7 Layer 3 VPN Configuration Guidelines 79

Enable a Signaling Protocol	81
Use LDP for VPN Signaling	81
Use RSVP for VPN Signaling	83
Configure an IGP on PE and Provider Routers	84
Configure an IBGP Session between PE Routers	85
Configure Routing Instances for Layer 3 VPNs on PE Routers	85
Configure the Description	86
Configure the Instance Type	86

Configure Interfaces for VPN Routing	86
Configure a Logical Unit on the Loopback Interface	88
Configure the Route Distinguisher	89
Configure Policy for the PE Router's VRF Table	89
Configure the Route Target	89
Configure the Route Origin	90
Configure Import Policy for the PE Router's VRF Table	90
Configure Export Policy for the PE Router's VRF Table	91
Apply Both the VRF Export and the BGP Export Policies	92
Configure a VRF Target	93
Filter Traffic Based on the IP Header	94
Egress Filtering Options and Limitations	95
Configure a VPN Tunnel for VRF Table Lookup	95
Configure VPN Routing between the PE and CE Routers	96
Configure BGP between the PE and CE Routers	96
Configure OSPF between the PE and CE Routers	96
Configure an OSPF Domain ID	97
Hub-and-Spoke Layer 3 VPNs and OSPF Domain ID	98
Compatibility with JUNOS Releases before 5.3	98
Example Configurations for Compatibility with JUNOS Releases before 5.3	99
Configure RIP between the PE and CE Routers	100
Configure Static Routes between the PE and CE Routers	101
Limit the Routes Accepted from a CE Router	101
Configure IPv6 between the PE and CE Routers	102
Configure IPv6 on the PE Router	102
Configure BGP or Static Routes on the PE Router	102
Configure BGP on the PE Router to Handle IPv6 Routes	103
Configure BGP on the PE Router to Handle IPv4 and IPv6 Routes	103
Configure Static Routes on the PE Router	104
Configure IPv6 on the Interfaces	104
Configure EBGp or IBGP Multihop between PE and CE Routers	105
Configure VPN Graceful Restart	105
Configure Multicast over Layer 3 VPNs	105
Configure a GRE Tunnel Interface for Layer 3 VPNs	106
Configure the GRE Tunnel Interface on the PE Router	106
Configure the GRE Tunnel Interface on the CE Router	107
Configure an ES Tunnel Interface for Layer 3 VPNs	108
Configure the ES Tunnel Interface on the PE Router	108
Configure the ES Tunnel Interface on the CE Router	109
Configure IPSec between PE Routers Instead of MPLS	110
Configure Packet Forwarding for VPNs	112

Chapter 8

Layer 3 VPN Configuration Troubleshooting Guidelines	113
--	-----

Diagnose Common Problems	113
Use the ping and traceroute Commands to	
Troubleshoot Layer 3 VPN Topologies	117
Ping One CE Router from the Other	118
Ping the Remote PE and CE Routers from the Local CE Router	119
Ping the Directly Connected PE and CE Routers from Each Other	121
Ping a Remote CE Router from a PE Router	123

Limitation on Pinging a Remote CE Router from a PE Router.....	124
Configure a Logical Unit on the Loopback Interface.....	124
Disable Normal TTL Decrementing for Layer 3 VPNs.....	125
Indirect Next-hop Address Space and Route Reflectors.....	125

Chapter 9

Layer 3 VPN Configuration Examples127

Configure a Simple Full-Mesh VPN Topology.....	128
Enable an IGP on the PE and Provider Routers.....	129
Enable RSVP and MPLS on the Provider Router.....	130
Configure the MPLS LSP Tunnel between the PE Routers	130
Configure IBGP on the PE Routers	131
Configure Routing Instances for VPNs on the PE Routers	132
Configure VPN Policy on the PE Routers.....	134
Simple VPN Configuration Summarized by Router	137
Router A (PE Router)	137
Router B (Provider Router)	139
Router C (PE Router)	140
Configure a Full-Mesh VPN Topology with Route Reflectors.....	142
Configure a Hub-and-Spoke VPN Topology	142
Enable an IGP on the Hub-and-Spoke PE Routers	145
Configure LDP on the Hub-and-Spoke PE Routers.....	145
Configure IBGP on the PE Routers	146
Configure Routing Instances for VPNs on the Hub-and-Spoke PE Routers.....	147
Configure VPN Policy on the PE Routers.....	150
Hub-and-Spoke VPN Configuration Summarized by Router	153
Router D (Hub PE Router).....	153
Router E (Spoke PE Router)	155
Router F (Spoke PE Router)	156
Configure an LDP-over-RSVP VPN Topology	158
Enable an IGP on the PE and Provider Routers.....	161
Enable LDP on the PE and Provider Routers.....	162
Enable RSVP and MPLS on the Provider Router.....	163
Configure the MPLS LSP Tunnel between the Provider Routers	163
Configure IBGP on the PE Routers	164
Configure Routing Instances for VPNs on the PE Routers	165
Configure VPN Policy on the PE Routers.....	166
LDP-over-MPLS VPN Configuration Summarized by Router	168
Router PE1	168
Router P1	169
Router P2	170
Router P3	170
Router PE2	171
Configure an Application-Based Layer 3 VPN Topology	172
Configuration on Router A	174
Configuration on Router E	176
Configuration for Router F	177
Configure an OSPF Domain ID for a Layer 3 VPN.....	177
Configure Interfaces on Router PE1	179
Configure Routing Options on Router PE1	179
Configure Protocols on Router PE1	180
Configure Policy Options on Router PE1.....	180

Configure the Routing Instance on Router PE1	181
Configuration Summary for Router PE1	182
Configure Overlapping VPNs Using Routing Table Groups	183
Configure Routing Table Groups	185
Configure Static Routes between the PE and CE Routers	185
Configure the Routing Instance for VPN A	186
Configure the Routing Instance for VPN AB	186
Configure the Routing Instance for VPN B	187
Configure VPN Policy	188
Configure BGP between the PE and CE Routers	190
Configure OSPF between the PE and CE Routers	192
Configure Static, BGP, and OSPF Routes between the PE and CE Routers	193
Configuring Overlapping VPNs Using Automatic Route Export	195
Configuring Overlapping VPNs with BGP and auto-export	196
Configuring Overlapping VPNs and Additional Tables	197
Configuring auto-export for all VRF Instances	198
Configure a GRE Tunnel Interface between PE Routers	198
Configure the Routing Instance on Router A	199
Configure the Routing Instance on Router D	199
Configure MPLS, BGP, and OSPF on Router A	200
Configure MPLS, BGP, and OSPF on Router D	200
Configure the Tunnel Interface on Router A	201
Configure the Tunnel Interface on Router D	201
Configure the Routing Options on Router A	201
Configure the Routing Options on Router D	202
Configuration Summary for Router A	202
Configuration Summary for Router D	204
Configure a GRE Tunnel Interface between a PE and CE Router	205
Configure the Routing Instance without the Encapsulating Interface	205
Configure the Routing Instance on Router PE1	206
Configure the GRE Tunnel Interface on Router PE1	206
Configure the Encapsulation Interface on Router PE1	206
Configure the Routing Instance with the Encapsulating Interface	207
Configure the Routing Instance on Router PE1	207
Configure the GRE Tunnel Interface on Router PE1	207
Configure the Encapsulation Interface on Router PE1	208
Configure the GRE Tunnel Interface on Router CE1	208
Configure an ES Tunnel Interface between a PE and CE Router	208
Configure IPSec on Router PE1	209
Configure the Routing Instance without the Encapsulating Interface	209
Configure the Routing Instance on Router PE1	210
Configure the ES Tunnel Interface on Router PE1	210
Configure the Encapsulating Interface for the ES Tunnel on Router PE1	210
Configure the Routing Instance with the Encapsulating Interface	210
Configure the Routing Instance on Router PE1	211
Configure the ES Tunnel Interface on Router PE1	211
Configure the Encapsulating Interface on Router PE1	212
Configure the ES Tunnel Interface on Router CE1	212
Configure IPSec on Router CE1	212
Configure SCU and DCU for Layer 3 VPNs	213

Non-VRF Internet Access	215
CE Router Accesses Internet Independently of the PE Router	215
PE Router Provides Layer 2 Internet Service.....	216
Distributed Internet Access.....	216
Route VPN and Internet Traffic through Different Interfaces.....	217
Configure Interfaces on Router PE1	219
Configure Routing Options on Router PE1.....	219
Configure BGP, IS-IS, and LDP Protocols on Router PE1	219
Configure a Routing Instance on Router PE1	220
Configure Policy Options on Router PE1	221
Traffic Routed by Different Interfaces Configuration	
Summarized by Router.....	222
Router PE1	222
Route VPN and Outgoing Internet Traffic through the Same Interface	
and Route Return Internet Traffic through a Different Interface	224
Configuration for Router PE1.....	224
Route VPN and Internet Traffic through the Same	
Interface Bidirectionally (VPN Has Public Addresses).....	225
Configure Routing Options on Router PE1.....	225
Configure Routing Protocols on Router PE1.....	226
Configure the Routing Instance on Router PE1	227
Traffic Routed through the Same Interface Bidirectionally	
Configuration Summarized by Router	227
Router PE1	227
Route VPN and Internet Traffic through the Same	
Interface Bidirectionally (VPN Has Private Addresses)	229
Configure Routing Options for Router PE1	229
Configure a Routing Instance for Router PE1.....	230
Configure Policy Options for Router PE1	231
Traffic Routed by the Same Interface Bidirectionally (VPN Has Private	
Addresses) Configuration Summarized by Router.....	231
Router PE1	231
Route Internet Traffic through a Separate NAT Device.....	233
Configure Interfaces on Router PE1	235
Configure Routing Options for Router PE1	235
Configure Routing Protocols on Router PE1.....	236
Configure a Routing Instance for Router PE1.....	236
Traffic Routed by Separate NAT Device	
Configuration Summarized by Router	238
Router PE1	238
Centralized Internet Access.....	241
Route Internet Traffic through a Hub CE Router	241
Configure a Routing Instance on Router PE1	242
Configure Policy Options on Router PE1	243
Internet Traffic Routed by a Hub CE Router	
Configuration Summarized by Router	244
Router PE1	244
Route Internet Traffic through Multiple CE Routers.....	245
Configure a Routing Instance on Router PE1	246
Configure Policy Options on Router PE1	247
Configure a Routing Instance on Router PE3	248
Configure Policy Options on Router PE3	248

Route Internet Traffic through Multiple CE Routers	
Configuration Summarized by Router	249
Router PE1	249
Router PE2	250
Router PE3	250

Chapter 11

Summary of Layer 3 VPN Configuration Statements	253
---	-----

description	253
inet6-vpn	253
instance-type	254
interface	254
route-distinguisher	255
route-distinguisher-id	255
vpn-apply-export	256
vrf-export	256
vrf-import	256
vrf-table-label	257
vrf-target	257

Part 4

Interprovider and Carrier-of-Carriers VPNs

Chapter 12

Interprovider and Carrier-of-Carriers VPNs Overview	261
---	-----

Interprovider and Carrier-of-Carriers VPN Standards	261
Traditional VPNs, Interprovider VPNs, and Carrier-of-Carriers VPNs	262
Standard VPNs	262
Interprovider and Carrier-of-Carriers VPNs	262
Interprovider VPNs	263
Interprovider VPNs—Linking VRF Tables between Autonomous Systems	264
Interprovider VPNs—Configure MP-EBGP between AS Border Routers	264
Interprovider VPNs—Configure Multihop MP-EBGP	265
Carrier-of-Carriers VPNs	265
Internet Service Provider as the Customer	267
VPN Service Provider as the Customer	267

Chapter 13

Interprovider and Carrier-of-Carriers Configuration Guidelines	269
--	-----

Interprovider VPNs	269
Interprovider VPNs Using MP-EBGP	270

Configure the AS Border Routers	270
Configure RSVP	270
Configure MPLS	270
Configure BGP	271
Configure OSPF	271
Interprovider VPNs Using Multihop MP-EBGP	272
Configure the AS Border Routers	272
Configure BGP	272
Configure Policy Options	273
Configure the PE Router	273
Carrier-of-Carriers VPNs	274
Carrier-of-Carriers VPN—Customer Provides Internet Service	274
Configure the Carrier-of-Carriers VPN Service Customer's CE Router ...	274
Configure MPLS	274
Configure BGP	275
Configure OSPF	275
Configure Policy Options	276
Configure the Carrier-of-Carriers VPN Service Provider's PE Routers ...	276
Configure MPLS	276
Configure BGP	276
Configure IS-IS	277
Configure LDP	277
Configure a Routing Instance	277
Configure Policy Options	278
Carrier-of-Carriers VPN—Customer Provides VPN Service	278
Configure the Carrier-of-Carriers Customer's PE Router	279
Configure MPLS	279
Configure BGP	279
Configure OSPF	280
Configure LDP	280
Configure VPN Service in the Routing Instance	280
Configure Policy Options	281
Configure the Carrier-of-Carriers Customer's CE Router	281
Configure MPLS	281
Configure BGP	282
Configure OSPF and LDP	282
Configure Policy Options	283
Configure the Provider's PE Router	283
Configure MPLS	283
Configure a PE-Router-to-PE-Router BGP Session	283
Configure IS-IS and LDP	284
Configure Policy Options	284
Configure a Routing Instance to Send Labeled Routes to the CE Router	285

Chapter 14

Configuration Examples for Interprovider and Carrier-of-Carriers VPNs

Example Terminology	287
Interprovider VPN Examples	288
Interprovider VPN Example—MP-EBGP between ISP Peer Routers	289
Configuration for Router A	289
Configuration for Router B	290

Configuration for Router C	291
Configuration for Router D	292
Configuration for Router E	293
Configuration for Router F	295
Interprovider VPN Example—Multihop MP-EBGP	295
Configuration for Router B	296
Configuration for Router C	296
Configuration for Router D	297
Configuration for Router E	298
Carrier-of-Carriers VPN Examples	298
Carrier-of-Carriers VPN Example—Customer Provides Internet Service	300
Configuration for Router A	300
Configuration for Router B	300
Configuration for Router C	301
Configuration for Router D	302
Configuration for Router E	303
Configuration for Router F	304
Configuration for Router G	305
Configuration for Router H	305
Configuration for Router I	307
Configuration for Router J	307
Configuration for Router K	308
Configuration for Router L	309
Carrier-of-Carriers VPN Example—Customer Provides VPN Service	309
Configuration for Router A	310
Configuration for Router B	310
Configuration for Router C	312
Configuration for Router D	312
Configuration for Router E	313
Configuration for Router F	315
Configuration for Router G	315
Configuration for Router H	315
Configuration for Router I	317
Configuration for Router J	318
Configuration for Router K	319
Configuration for Router L	320
Multiple Instances for LDP and Carrier-of-Carriers VPNs	320

Chapter 15

Summary of the Interprovider and Carrier-of-Carriers Configuration Statement	321
---	-----

labeled-unicast	321
-----------------------	-----

Part 5

Layer 2 Circuits

Chapter 16

Layer 2 Virtual Circuit Overview	325
--	-----

Chapter 17

Layer 2 Circuit Configuration.....	327
------------------------------------	-----

Configure the Neighbor and Interface	327
Configure the Virtual Circuit ID	328
Configure the Interface Encapsulation Type.....	328
Configure LDP for Layer 2 Circuits.....	328
Disable the Control Word for Layer 2 Circuits	329
Trace Layer 2 Circuit Creation and Changes	329

Chapter 18

Summary of Layer 2 Circuit Configuration Statements	331
---	-----

control-word	331
interface	331
neighbor	332
no-control-word	332
traceoptions	332
virtual-circuit-id	334

Part 6

Appendix

Appendix A

Glossary.....	337
---------------	-----

Part 7

Index

Index

Index.....	359
------------	-----

Index

Index of Statements and Commands.....	363
---------------------------------------	-----

List of Figures

List of Figures

Figure 1:	VPN Router Components	4
Figure 2:	Layer 2 VPN Connecting CE Routers	10
Figure 3:	Example of a Simple Full-Mesh Layer 2 VPN Topology	30
Figure 4:	VPN Attributes and Route Distribution	61
Figure 5:	Overlapping Addresses among Different VPNs	62
Figure 6:	Route Distinguishers	64
Figure 7:	VRF Tables	65
Figure 8:	Route Distribution within a VPN	68
Figure 9:	Distribution of Routes from CE Routers to PE Routers	69
Figure 10:	Distribution of Routes between PE Routers	70
Figure 11:	Distribution of Routes from PE Routers to CE Routers	71
Figure 12:	Using MPLS LSPs to Tunnel between PE Routers	72
Figure 13:	Label Stack	73
Figure 14:	Multicast Topology Overview	75
Figure 15:	Layer 3 VPN Topology for ping and traceroute Command Examples	117
Figure 16:	Example of a Simple VPN Topology	128
Figure 17:	Example of a Hub-and-Spoke VPN Topology	143
Figure 18:	Route Distribution between Two Spoke Routers	144
Figure 19:	Example of an LDP-over-RSVP VPN Topology	158
Figure 20:	Label Pushing and Popping	160
Figure 21:	Application-Based Layer 3 VPN Example Configuration	174
Figure 22:	Example of a Configuration Using an OSPF Domain ID	178
Figure 23:	Example of an Overlapping VPN Topology	184
Figure 24:	PE Router A and PE Router D Connected by a GRE Tunnel Interface	198
Figure 25:	GRE Tunnel between the CE Router and the PE Router	205
Figure 26:	ES Tunnel Interface (IPSec Tunnel) between the CE Router and the PE Router	208
Figure 27:	PE Router Does Not Provide Internet Access	216
Figure 28:	PE Router Connects to a Router Connected to the Internet	216
Figure 29:	Routing VPN and Internet Traffic through Different Interfaces	217
Figure 30:	Example of Internet Traffic Routed through Separate Interfaces	218
Figure 31:	VPN and Outgoing Internet Traffic Routed through the Same Interface and Return Internet Traffic Routed through a Different Interface	224
Figure 32:	Interface Configured to Carry Both Internet and VPN Traffic	225
Figure 33:	VPN and Internet Traffic Routed through the Same Interface	229
Figure 34:	Overview of Internet Traffic Routed through a Separate NAT Device	233
Figure 35:	Internet Traffic Routed through a Separate NAT Device Example Topology	234
Figure 36:	Internet Access through a Hub CE Router Performing NAT	241
Figure 37:	Internet Access Provided through a Hub CE Router	242

List of Figures

Figure 38: Two Hub CE Routers Handling Internet Traffic and NAT	246
Figure 39: Interprovider VPN Network Topology	263
Figure 40: Carrier-of-Carriers VPN Architecture	266
Figure 41: Network Topology of Interprovider VPN Examples	288
Figure 42: Carrier-of-Carriers VPN Example Network Topology	299
Figure 43: Components of a Layer 2 Virtual Circuit	325

About This Manual

This chapter provides a high-level overview of the *JUNOS Internet Software Configuration Guide: VPNs*:

- Objectives on page xix
- Audience on page xx
- Document Organization on page xx
- Part Organization on page xxi
- Using the Indexes on page xxii
- Documentation Conventions on page xxiii
- List of Technical Publications on page xxv
- Documentation Feedback on page xxv
- How to Request Support on page xxvi

Objectives

This manual provides an overview of the JUNOS Internet software virtual private network (VPN) functions and describes how to configure VPNs on the router.

This manual documents Release 5.6 of the JUNOS Internet software. To obtain additional information about the JUNOS software—either corrections to information in this manual or information that might have been omitted from this manual—refer to the software release notes.

To obtain the most current version of this manual and the most current version of the software release notes, refer to the product documentation page on the Juniper Networks Web site, which is located at <http://www.juniper.net/>.

To order printed copies of this manual or to order a documentation CD-ROM, which contains this manual, please contact your sales representative.

Audience

This manual is designed for network administrators who are configuring a Juniper Networks router. It assumes that you have a broad understanding of networks in general, the Internet in particular, networking principles, and network configuration. This manual assumes that you are familiar with one or more of the following Internet routing protocols: Border Gateway Protocol (BGP), Routing Information Protocol (RIP), Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First (OSPF), Internet Control Message Protocol (ICMP) router discovery, Internet Group Management Protocol (IGMP), Distance Vector Multicast Routing Protocol (DVMRP), Protocol-Independent Multicast (PIM), Multiprotocol Label Switching (MPLS), Resource Reservation Protocol (RSVP), and Simple Network Management Protocol (SNMP).

Document Organization

This manual is divided into several parts. Each part describes a major functional area of the JUNOS software, and the individual chapters within a part describe the software commands of that functional area.

This manual contains the following parts and chapters:

- Preface, “About This Manual” (this chapter), provides a brief description of the contents and organization of this manual and describes how to contact customer support.
- Part 1, “VPN Overview,” provides an overview of Layer 2 and Layer 3 VPNs.
 - Chapter 1, “VPN Overview,” provides common VPN terminology and describes the major differences between Layer 2 and Layer 3 VPNs.
- Part 2, “Layer 2 VPNs,” describes how to configure the JUNOS software to support Layer 2 VPNs.
 - Chapter 2, “Layer 2 VPN Overview,” provides an overview of Layer 2 VPNs.
 - Chapter 3, “Layer 2 VPN Configuration Guidelines,” describes the minimum and optional configurations for Layer 2 VPNs.
 - Chapter 4, “Layer 2 VPN Configuration Example,” provides a configuration example for a Layer 2 VPN.
 - Chapter 5, “Summary of Layer 2 VPN Configuration Statements,” describes the statements used to configure Layer 2 VPNs.
- Part 3, “Layer 3 VPNs,” describes how to configure the JUNOS software to support Layer 3 VPNs.
 - Chapter 6, “Layer 3 VPN Overview,” provides an overview of Layer 3 VPNs.
 - Chapter 7, “Layer 3 VPN Configuration Guidelines,” describes the minimum and optional configurations for Layer 3 VPNs.
 - Chapter 8, “Layer 3 VPN Configuration Troubleshooting Guidelines,” provides guidance for troubleshooting Layer 3 VPNs.
 - Chapter 9, “Layer 3 VPN Configuration Examples,” provides configuration examples for Layer 3 VPNs.

- Chapter 10, “Layer 3 VPN Internet Access Examples,” provides configuration examples that show how to configure Internet access for Layer 3 VPNs.
- Chapter 11, “Summary of Layer 3 VPN Configuration Statements,” describes the statements used to configure Layer 3 VPNs.
- Part 4, “Interprovider and Carrier-of-Carriers VPNs,” describes how to configure the JUNOS software to support interprovider and carrier-of-carriers VPNs.
 - Chapter 12, “Interprovider and Carrier-of-Carriers VPNs Overview,” provides an overview of interprovider and carrier-of-carriers VPNs.
 - Chapter 13, “Interprovider and Carrier-of-Carriers Configuration Guidelines,” describes the minimum and optional configurations for interprovider and carrier-of-carriers VPNs.
 - Chapter 14, “Configuration Examples for Interprovider and Carrier-of-Carriers VPNs,” provides examples that show how to configure interprovider and carrier-of-carriers VPNs.
 - Chapter 15, “Summary of the Interprovider and Carrier-of-Carriers Configuration Statement,” describes the statement used to configure interprovider and carrier-of-carriers VPNs.
- Part 5, “Layer 2 Circuits,” describes how to configure Layer 2 circuits.
 - Chapter 16, “Layer 2 Virtual Circuit Overview,” provides an overview of Layer 2 virtual circuits.
 - Chapter 17, “Layer 2 Circuit Configuration,” describes how to configure Layer 2 virtual circuits.
 - Chapter 18, “Summary of Layer 2 Circuit Configuration Statements,” describes the statements used to configure Layer 2 circuits.

This manual also contains a glossary, a complete index, and an index of statements and commands.

Part Organization

The parts in this manual typically contain the following chapters:

- Overview—Provides background information about and discusses concepts related to the software component described in that part of the book.
- Configuration statements—Lists all the configuration statements available to configure the software component. This list is designed to provide an overview of the configuration statement hierarchy for that software component.
- Configuration guidelines—Describes how to configure all the features of the software component. The first section of the configuration guidelines describes the minimum configuration for that component, listing the configuration statements you must include to enable the software component on the router with only the bare minimum functionality. The remaining sections in the configuration guidelines are generally arranged so that the most common features are near the beginning.

- **Statement summary**—A reference that lists all configuration statements alphabetically and explains each statement and all its options. The explanation of each configuration statement consists of the following parts:
 - **Syntax**—Describes the full syntax of the configuration statement. For an explanation of how to read the syntax statements, see “Documentation Conventions” on page xxiii.
 - **Hierarchy level**—Tells where in the configuration statement hierarchy you include the statement.
 - **Description**—Describes the function of the configuration statement.
 - **Options**—Describes the configuration statement’s options, if there are any. For options with numeric values, the allowed range and default value, if any, are listed. For multiple options, if one option is the default, that fact is stated. If a configuration statement is at the top of a hierarchy of options that are other configuration statements, these options are generally explained separately in the statement summary section.
 - **Usage guidelines**—Points to the section or sections in the configuration guidelines section that describe how to use the configuration statement.
 - **Required privilege level**—Indicates the permissions that the user must have to view or modify the statement in the router configuration. For an explanation of the permissions, see the *JUNOS Internet Software Configuration Guide: Getting Started*.
 - **See also**—Indicates other configuration statements that might provide related or similar functionality.

Using the Indexes

This manual contains two indexes: a complete index, which contains all index entries, and an index that contains only statements and commands.

In the complete index, bold page numbers point to pages in the statement summary chapters. The index entry for each configuration statement always contains at least two entries. The first, with a bold page number on the same line as the statement name, references the statement summary section. The second entry, “usage guidelines,” references the section in a configuration guidelines chapter that describes how to use the statement.

Documentation Conventions

General Conventions

This manual uses the following text conventions:

- Statements, commands, filenames, directory names, IP addresses, and configuration hierarchy levels are shown in a sans serif font. In the following example, *stub* is a statement name and [edit protocols ospf area *area-id*] is a configuration hierarchy level:

To configure a stub area, include the stub statement at the [edit protocols ospf area *area-id*] hierarchy level:

- In examples, text that you type literally is shown in bold. In the following example, you type the word *show*:

```
[edit protocols ospf area area-id]  
cli# show  
stub <default-metric metric>
```

- Examples of command output are generally shown in a fixed-width font to preserve the column alignment. For example:

```
> show interfaces terse  
Interface      Admin Link Proto Local              Remote  
at-1/3/0       up    up  
at-1/3/0.0     up    up    inet  1.0.0.1            --> 1.0.0.2  
                                     iso  
fxp0           up    up  
fxp0.0         up    up    inet  192.168.5.59/24
```

Conventions for Software Commands and Statements

When describing the JUNOS software, this manual uses the following type and presentation conventions:

- Statement or command names that you type literally are shown nonitalicized. In the following example, the statement name is *area*:

You configure all these routers by including the following area statement at the [edit protocols ospf] hierarchy level:

- Options, which are variable terms for which you substitute appropriate values, are shown in italics. In the following example, *area-id* is an option. When you type the area statement, you substitute a value for *area-id*.

```
area area-id;
```

- Optional portions of a configuration statement are enclosed in angle brackets. In the following example, the “default-metric *metric*” portion of the statement is optional:

```
stub <default-metric metric>;
```

- For text strings separated by a pipe (|), you must specify either *string1* or *string2*, but you cannot specify both or neither of them. Parentheses are sometimes used to group the strings.

```
string1 | string2
(string1 | string2)
```

In the following example, you must specify either broadcast or multicast, but you cannot specify both:

```
broadcast | multicast
```

- For some statements, you can specify a set of values. The set must be enclosed in square brackets. For example:

```
community name members [ community-ids ]
```

- The configuration examples in this manual are generally formatted in the way that they appear when you issue a show command. This format includes braces ({ }) and semicolons. When you type configuration statements in the CLI, you do not type the braces and semicolons. However, when you type configuration statements in an ASCII file, you must include the braces and semicolons. For example:

```
[edit]
cli# set routing-options static route default nexthop address retain
[edit]
cli# show
routing-options {
  static {
    route default {
      nexthop address;
      retain;
    }
  }
}
```

- Comments in the configuration examples are shown either preceding the lines that the comments apply to, or more often, they appear on the same line. When comments appear on the same line, they are preceded by a pound sign (#) to indicate where the comment starts. In an actual configuration, comments can only precede a line; they cannot be on the same line as a configuration statement. For example:

```
protocols {
  mpls {
    interface (interface-name | all); # Required to enable MPLS on the interface
  }
  rsvp {
    interface interface-name; # Required for dynamic MPLS only
  }
}
```

- The general syntax descriptions provide no indication of the number of times you can specify a statement, option, or keyword. This information is provided in the text of the statement summary.

List of Technical Publications

Table 1 lists the software and hardware books for Juniper Networks routers and describes the contents of each book.

Table 1: Juniper Networks Technical Documentation

Book	Description
JUNOS Internet Software Configuration Guides	
<i>Getting Started</i>	Provides an overview of the JUNOS Internet software and describes how to install and upgrade the software. This manual also describes how to configure system management functions and how to configure the chassis, including user accounts, passwords, and redundancy.
<i>Interfaces and Class of Service</i>	Provides an overview of the interface and class-of-service functions of the JUNOS Internet software and describes how to configure the interfaces on the router.
<i>MPLS Applications</i>	Provides an overview of traffic engineering concepts and describes how to configure traffic engineering protocols.
<i>Multicast</i>	Provides an overview of multicast concepts and describes how to configure multicast routing protocols.
<i>Network Management</i>	Provides an overview of network management concepts and describes how to configure various network management features, such as SNMP, accounting options, and cflowd.
<i>Policy Framework</i>	Provides an overview of policy concepts and describes how to configure routing policy, firewall filters, and forwarding options.
<i>Routing and Routing Protocols</i>	Provides an overview of routing concepts and describes how to configure routing, routing instances, and unicast routing protocols.
<i>VPNs</i>	Provides an overview of Layer 2 and Layer 3 Virtual Private Networks (VPNs), describes how to configure VPNs, and provides configuration examples.
JUNOS Internet Software References	
<i>Operational Mode Command Reference: Interfaces</i>	Describes the JUNOS Internet software operational mode commands you use to monitor and troubleshoot interfaces on Juniper Networks M-series and T-series routers.
<i>Operational Mode Command Reference: Protocols, Class of Service, Chassis, and Management</i>	Describes the JUNOS Internet software operational mode commands you use to monitor and troubleshoot most aspects of Juniper Networks M-series and T-series routers.
<i>System Log Messages Reference</i>	Describes how to access and interpret system log messages generated by JUNOS software modules and provides a reference page for each message.
JUNOScript API Documentation	
<i>JUNOScript API Guide</i>	Describes how to use the JUNOScript API to monitor and configure Juniper Networks routers.
<i>JUNOScript API Reference</i>	Provides a reference page for each tag in the JUNOScript API.
JUNOS Internet Software Comprehensive Index	
<i>Comprehensive Index</i>	Provides a complete index of all JUNOS Internet software books and the <i>JUNOScript API Guide</i> .
Hardware Documentation	
<i>Hardware Guide</i>	Describes how to install, maintain, and troubleshoot routers and router components. Each router platform (M5 and M10 routers, M20 router, M40 router, M40e router, M160 router, T320 router, and T640 routing node) has its own hardware guide.
<i>PIC Guide</i>	Describes the router Physical Interface Cards (PICs). Each router platform has its own PIC guide.

Documentation Feedback

We are always interested in hearing from our customers. Please let us know what you like and do not like about the Juniper Networks documentation, and let us know of any suggestions you have for improving the documentation. Also, let us know if you find any mistakes in the documentation. Send your feedback to techpubs-comments@juniper.net.

How to Request Support

For technical support, contact Juniper Networks at support@juniper.net, or at 1-888-314-JTAC (within the United States) or 408-745-2121 (from outside the United States).

Part 1

VPN Overview

■ VPN Overview on page 3

.....

Chapter 1

VPN Overview

A virtual private network (VPN) consists of two topological areas, the provider's network and the customer's network. The provider's network, which runs across the public Internet infrastructure, consists of routers that provide VPN services to a customer's network as well as routers that provide other services. The customer's network is commonly located at multiple physical sites. The provider's network acts to connect the various customer sites in what appears to the customer and the provider to be a private network.

To ensure that VPNs remain private and isolated from other VPNs and from the public Internet, the provider's network maintains policies that keep routing information from different VPNs separate.

A provider can service multiple VPNs as long as its policies keep routes from different VPNs separate. Similarly, a site can belong to multiple VPNs as long as it keeps routes from the different VPNs separate.

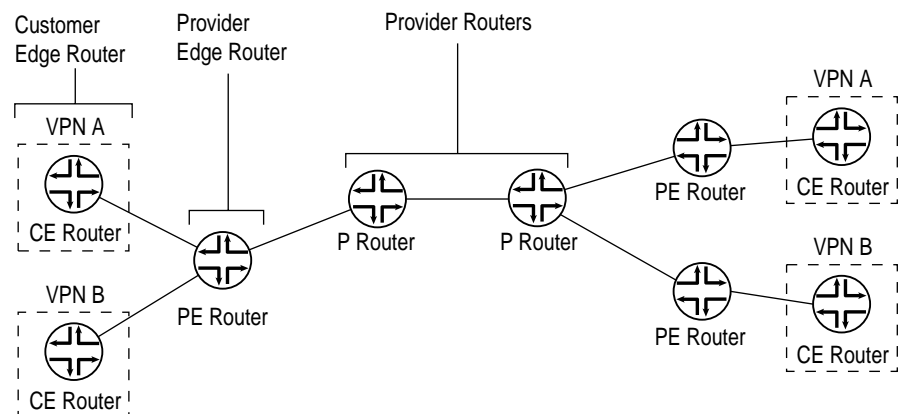
VPN Terminology

VPNs contain the following types of network devices (see Figure 1):

- **Provider edge (PE) routers**—Routers in the provider's network that connect to customer edge devices located at customer sites. PE routers support VPN and label functionality. (The label functionality can be provided either by the Resource Reservation Protocol [RSVP] or Label Distribution Protocol [LDP].) Within a single VPN, pairs of PE routers are connected through a tunnel, which can be either an Multiprotocol Label Switching (MPLS) label switched path (LSP) or an LDP tunnel.
- **Provider (P) routers**—Routers within the core of the provider's network that are not connected to any routers at a customer site but are part of the tunnel between pairs of PE routers. Provider routers support MPLS LSP or LDP functionality, but do not need to support VPN functionality.
- **Customer edge (CE) devices**—Routers or switches located at the customer's site that connect to the provider's network. CE devices are typically IP routers.

VPN functionality is provided by the PE routers; the provider and CE routers have no special configuration requirements for VPNs.

Figure 1: VPN Router Components



1664

Differences between Layer 2 and Layer 3 VPNs

In a Layer 3 VPN, the routing occurs on the service provider's routers. In a Layer 2 VPN, routing occurs on the customer's routers, typically on the CE router. Layer 3 VPNs require more configuration on the part of the service provider, because the service provider's PE routers must know the customer's routes. Layer 2 VPNs require less configuration on the part of the service provider, because routing is handled by the customer's routers and not the service provider's.

VPN Graceful Restart

VPN graceful restart allows a router whose VPN control plane is undergoing a restart to continue to forward traffic while recovering its state from neighboring routers. Without graceful restart, a control plane restart disrupts any Layer 2 or Layer 3 VPN services provided by the router.

For VPN graceful restart to function properly, the following needs to be configured on the PE router:

- BGP graceful restart must be active on the PE-to-PE sessions carrying any service signaling data in the session's network layer reachability information (NLRI).
- OSPF, ISIS, LDP, and RSVP graceful restart must be active, because routes added by these protocols are used to resolve Layer 2 and Layer 3 VPN NLRI.
- For other protocols (static, RIP, OSPF, LDP and so on), graceful restart functionality must also be active when these protocols are run between the PE and CE routers. Layer 2 VPNs do not rely on this because protocols are not configured between the PE and CE routers.

In VPN graceful restart, a restarting router does the following:

- Waits for all the BGP NLRI information from other PE routers before it starts advertising routes to its CE routers.
- Waits for all protocols in all routing instances to converge (or finish graceful restart) before sending CE router information to the other PE routers.
- Waits for all routing instance information (whether it is local configuration or advertisements from a remote peer router) to be processed before sending it to the other PE routers.
- Preserves all forwarding state information in the MPLS routing tables until new labels and transit routes are allocated and then advertises them to other PE routers (and CE routers in carrier-of-carriers VPNs).

Part 2

Layer 2 VPNs

- Layer 2 VPN Overview on page 9
- Layer 2 VPN Configuration Guidelines on page 11
- Layer 2 VPN Configuration Example on page 29
- Summary of Layer 2 VPN Configuration Statements on page 47

.....

Chapter 2

Layer 2 VPN Overview

This chapter provides an overview of Layer 2 Multiprotocol Label Switching (MPLS) virtual private networks (VPNs) as they are implemented in the JUNOS software.

For information about VPNs as they are implemented in the JUNOS software and information on the differences between Layer 2 and Layer 3 VPNs, see “VPN Overview” on page 3.

This chapter discusses the following topics:

- Layer 2 VPN Overview on page 9
- Layer 2 VPN Standards on page 10

Layer 2 VPN Overview

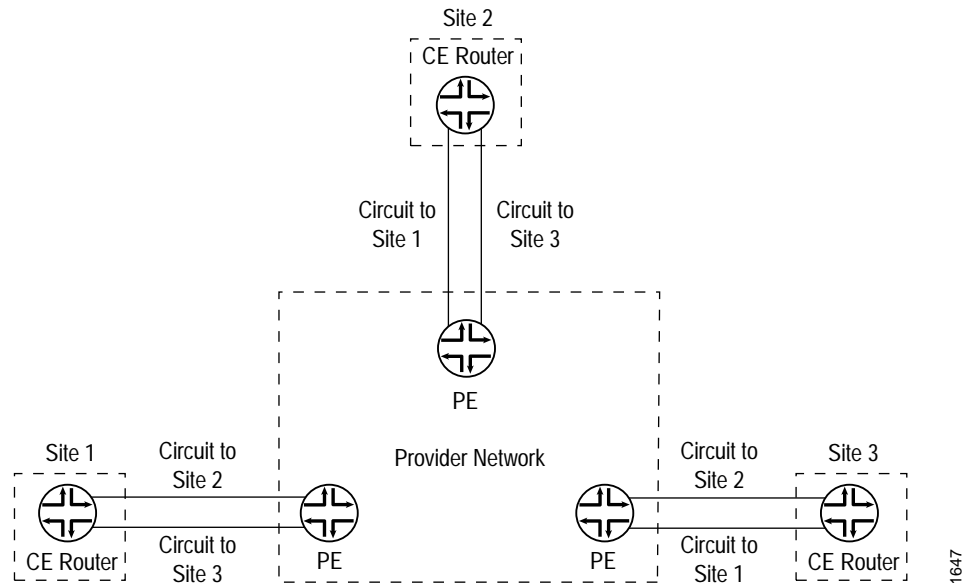
Implementing a Layer 2 VPN on a router is similar to implementing a VPN using a Layer 2 technology such as Asynchronous Transfer Mode (ATM) or Frame Relay. However, for a Layer 2 VPN on a router, traffic is forwarded to the router in a Layer 2 format. It is carried by MPLS over the service provider’s network, and then converted back to Layer 2 format at the receiving site. You can configure different Layer 2 formats at the sending and receiving sites. The security and privacy of an MPLS Layer 2 VPN are equal to those of an ATM or Frame Relay VPN.

On a Layer 2 VPN, routing occurs on the customer’s routers, typically on the customer edge (CE) router. The CE router connected to a service provider on a Layer 2 VPN must select the appropriate circuit on which to send traffic. The provider edge (PE) router receiving the traffic sends it across the service provider’s network to the PE router connected to the receiving site. PE routers do not need to know the customer’s routes or routing topology; they need to know only in which tunnel to send the data.

For a Layer 2 VPN, customers need to configure their own routers to carry all Layer 3 traffic. The service provider needs to know only how much traffic the Layer 2 VPN will need to carry. The service provider’s routers carry traffic between the customer’s sites using Layer 2 VPN interfaces. The VPN topology is determined by policies configured on the PE routers.

Customers need to know only which VPN interfaces connect to which of their own sites. Figure 2 illustrates a Layer 2 VPN in which each site has a VPN interface linked to each of the other customer sites.

Figure 2: Layer 2 VPN Connecting CE Routers



The benefits of implementing a Layer 2 MPLS VPN include:

- Service providers do not have to invest in separate Layer 2 equipment to provide Layer 2 VPN service. A Layer 2 MPLS VPN allows you to provide Layer 2 VPN service over an existing IP and MPLS backbone.
- You can configure the PE router to run any Layer 3 protocol in addition to the Layer 2 protocols.
- Customers who prefer to maintain control over most of the administration of their own networks might want Layer 2 VPN connections with their service provider instead of a Layer 3 VPN.

Layer 2 VPN Standards

For more information about Layer 2 VPNs, see *MPLS-based Layer 2 VPNs*, Internet draft draft-kompella-ppvpn-l2vpn-02.txt.

You can access Internet Request for Comments (RFCs) and drafts from the Internet Engineering Task Force (IETF) Web site at <http://www.ietf.org>.

Chapter 3

Layer 2 VPN Configuration Guidelines

To configure Layer 2 virtual private network (VPN) functionality, you must enable Layer 2 VPN support on the provider edge (PE) router. You must also configure PE routers to distribute routing information to the other PE routers in the VPN and configure the circuits between the PE routers and the customer edge (CE) routers.

Each Layer 2 VPN is configured under a routing instance of type l2vpn. An l2vpn routing instance can transparently carry Layer 3 traffic across the service provider's network. As with other routing instances, all logical interfaces belonging to a Layer 2 VPN routing instance are listed under that instance.

The configuration of the CE routers is not relevant to the service provider. The CE routers only need to provide appropriate Layer 2 circuits (with appropriate circuit identifiers, such as data-link connection identifier [DLCI], virtual path identifier/virtual channel identifier [VPI/VCI], or virtual local area network identifier [VLAN ID]) to send traffic to the PE router.

To configure Layer 2 VPNs, you include statements at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit]
routing-instances {
  routing-instance-name {
    description text;
    instance-type l2vpn;
    interface interface-name;
    route-distinguisher (as-number:id | ip-address:id);
    vrf-export [ policy-name ];
    vrf-import [ policy-name ];
    protocols {
      l2vpn {
        (control-word | no-control-word);
        encapsulation type;
        traceoptions {
          file filename <replace> <size size> <files number> <nostamp>;
          flag flag <flag-modifier> <disable>;
        }
        site site-name {
          site-identifier identifier;
          interface interface-name {
            remote-site-id remote-site-id;
          }
        }
      }
    }
  }
}
```

For Layer 2 VPNs, only some of the statements in the [edit routing-instances] hierarchy are valid. For the full hierarchy, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

In addition to these statements, you must configure Multiprotocol Label Switching (MPLS) label-switched paths (LSPs) between the PE routers, internal Border Gateway Protocol (IBGP) sessions between the PE routers, and an interior gateway protocol (IGP) on the PE and provider routers.

By default, Layer 2 VPNs are disabled.

This chapter describes the following tasks for configuring Layer 2 VPNs:

- Configure MPLS LSPs between the PE Routers on page 12
- Configure an IGP on the PE Routers on page 15
- Configure an IBGP Session between PE Routers on page 16
- Configure Routing Instances for Layer 2 VPNs on the PE Routers on page 16
- Configure the Connections to the Local Site on page 23
- Configure Packet Forwarding for VPNs on page 26
- Configure VPN Graceful Restart on page 27

Configure MPLS LSPs between the PE Routers

For Layer 2 VPNs to function, you must configure MPLS LSPs between the PE routers. You can do one of the following:

- Configure MPLS LSPs Using LDP on page 12
- Configure MPLS LSPs Using RSVP on page 14

Configure MPLS LSPs Using LDP

To use the Label Distribution Protocol (LDP) to configure the MPLS LSPs, perform the following steps on the PE and provider routers:

1. Configure LDP on the interfaces in the core of the service provider's network by including the `ldp` statement at the [edit protocols] hierarchy level. You need to configure LDP only on the interfaces between PE routers or between PE and provider routers. You can think of these as the "core-facing" interfaces.

```
[edit]
protocols {
  ldp {
    interface interface-name;
  }
}
```

2. Configure the MPLS address family on the interfaces on which you enabled LDP (the interfaces you configured in Step 1):

```
[edit]
interfaces {
  interface-name {
    unit logical-unit-number {
      family mpls;
    }
  }
}
```

Specify the interface name in the format *type-fpc/pic/port*.

3. Configure Open Shortest Path First (OSPF) or IS-IS on each PE and provider router. You configure these protocols at the master instance of the routing protocol, not within the routing instance used for the VPN.

To configure OSPF, include the `ospf` statement at the `[edit protocols]` hierarchy level. At a minimum, you must configure a backbone area on at least one of the router's interfaces.

```
[edit]
protocols {
  ospf {
    area 0.0.0.0 {
      interface interface-name;
    }
  }
}
```

To configure IS-IS, include the `isis` statement at the `[edit protocols]` hierarchy level and configure the loopback interface and International Organization for Standardization (ISO) family at the `[edit interfaces]` hierarchy level. At a minimum, you must enable IS-IS on the router, configure a network entity title (NET) on one of the router's interfaces (preferably the loopback interface, `lo0`), and configure the ISO family on all interfaces on which you want IS-IS to run. When you enable IS-IS, Level 1 and Level 2 are enabled by default. The following is the minimum IS-IS configuration. In the address statement, *address* is the NET.

```
[edit]
interfaces {
  lo0 {
    unit logical-unit-number {
      family iso {
        address address;
      }
    }
  }
  type-fpc/pic/port {
    unit logical-unit-number {
      family iso;
    }
  }
}
protocols {
  isis {
    interface all;
  }
}
```

For detailed information about how to configure LDP, see the *JUNOS Internet Software Configuration Guide: MPLS Applications*. For more information about configuring OSPF and IS-IS, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

Configure MPLS LSPs Using RSVP

To configure the MPLS LSPs using the Resource Reservation Protocol (RSVP), perform the following steps:

1. On each PE router, configure traffic engineering. To do this, you must configure an IGP that supports traffic engineering (either IS-IS or OSPF) and enable traffic engineering support for that protocol.

To enable OSPF traffic engineering support, include the `traffic-engineering` statement at the `[edit protocols ospf]` hierarchy level:

```
[edit protocols ospf]
  traffic-engineering;
```

For IS-IS, traffic engineering support is enabled by default.

2. On each PE and provider router, enable RSVP on the router interfaces that participate in the LSP. On the PE router, these are the interfaces that are the ingress and egress points to the LSP. On the provider router, these are the interfaces that connect the LSP between the PE routers.

To configure RSVP on the PE and provider routers, include the `interface` statement at the `[edit rsvp]` hierarchy level. Include one `interface` statement for each interface on which you are enabling RSVP.

```
[edit]
  rsvp {
    interface interface-name;
    interface interface-name;
  }
```

3. On each PE router, configure an MPLS LSP to the PE router that is the LSP's egress point. To do this, include the `label-switched-path` and `interface` statements at the `[edit mpls]` hierarchy level.

```
[edit]
  mpls {
    label-switched-path lsp-path-name {
      to ip-address;
    }
    interface interface-name;
  }
```

In the `to` statement, specify the address of the LSP's egress point, which is an address on the remote PE router.

In the `interface` statement, specify the name of the interface (both the physical and logical portions). Include one `interface` statement for the interface associated with the LSP.

When you configure the same interface at the [edit interfaces] hierarchy level, you must also configure family mpls and family inet when configuring the logical interface:

```
[edit interfaces]
interface-name {
  unit logical-unit-number {
    family inet;
    family mpls;
  }
}
```

4. On all provider routers that participate in the LSP, enable MPLS by including the interface statement at the [edit mpls] hierarchy level. Include one interface statement for each connection to the LSP.

```
[edit]
mpls {
  interface interface-name;
  interface interface-name;
}
```

5. Enable MPLS on the interface between the PE and CE routers by including the interface statement at the [edit mpls] hierarchy level. Doing this allows the PE router to assign an MPLS label to traffic entering the LSP or to remove the label from traffic exiting the LSP.

```
[edit]
mpls {
  interface interface-name;
}
```

For information about configuring RSVP or MPLS, see the *JUNOS Internet Software Configuration Guide: MPLS Applications*.

Configure an IGP on the PE Routers

To allow PE routers to exchange routing information, you must configure either an IGP or static routes on these routers. You configure the IGP on the master instance of the routing protocol process at the [edit protocols] hierarchy level, not within the routing instance used for the Layer 2 VPN—that is, not at the [edit routing-instances] hierarchy level.

When you configure the PE router, do not configure any summarization of the PE router's loopback addresses at the area boundary. Each PE router's loopback address should appear as a separate route.

For information about configuring routing protocols and static routes, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

Configure an IBGP Session between PE Routers

You must configure an IBGP session between the PE routers to allow these routers to exchange information about a Layer 2 VPN, particularly information about the sites connected to a Layer 2 VPN. The PE routers rely on this information to determine which labels to use for traffic destined for remote sites.

Configure an IBGP session for the Layer 2 VPN at the [edit protocols bgp group *group-name*] hierarchy level as follows:

```
[edit protocols]
bgp {
  group group-name {
    type internal;
    local-address ip-address;
    family l2vpn {
      unicast;
    }
    neighbor ip-address;
  }
}
```

The family l2vpn statement indicates that the IBGP session is for the Layer 2 VPN.

The IP address in the local-address statement is the address of the loopback interface (lo0) on the local PE router. The IBGP session for the Layer 2 VPN runs through the loopback address. (You must also configure the lo0 interface at the [edit interfaces] hierarchy level.)

The IP address in the neighbor statement is the loopback address of the neighboring PE router. If you are using RSVP signaling, this IP address is the same address you specify in the to statement at the [edit mpls label-switched-path *lsp-path-name*] hierarchy level when you configure the MPLS LSP.

Configure Routing Instances for Layer 2 VPNs on the PE Routers

To configure routing instances for Layer 2 VPNs, include the routing-instances statement at the [edit] hierarchy level. You configure Layer 2 VPN routing instances only on the PE routers.

Configure the routing instance as follows:

```
[edit]
routing-instances {
  routing-instance-name {
    description text;
    instance-type l2vpn;
    interface interface-name;
    control-word;
    route-distinguisher (as-number:id | ip-address:id);
    vrf-export [ policy-name ]
    vrf-import [ policy-name ]
  }
}
```




For the Layer 2 VPN to function, you must include the instance-type, interface, route-distinguisher, vrf-export, and vrf-import statements in the routing instance configuration on the PE router.

Note

The following sections describe how to configure Layer 2 VPN routing instances:

- Configure the Description on page 17
- Configure the Instance Type on page 17
- Configure Interfaces for Layer 2 VPN Routing on page 18
- Configure CCC Encapsulation on Interfaces on page 18
- Configure TCC Encapsulation on Interfaces on page 19
- Configure Layer 2 VPN Policing on Interfaces on page 20
- Configure the Route Distinguisher on page 21
- Disable the Control Word for Layer 2 VPNs on page 22
- Configure Policy for the Layer 2 VPN Routing Table on page 22

Configure the Description

To provide a textual description for the routing instance, include the description statement at the [edit routing-instances *routing-instance-name*] hierarchy level. Enclose any descriptive text that includes spaces in quotation marks (" "). Any descriptive text you include is displayed in the output of the show route instance detail command and has no effect on the operation of the routing instance.

```
[edit routing-instances routing-instance-name]
description text;
```

Configure the Instance Type

To enable Layer 2 VPN routing on a PE router, include the instance-type statement at the [edit routing-instances *routing-instance-name*] hierarchy level, specifying the instance type as l2vpn:

```
[edit routing-instances routing-instance-name]
instance-type l2vpn;
```

Configure Interfaces for Layer 2 VPN Routing

On each PE router, you must configure the interfaces over which the Layer 2 VPN traffic travels between PE and CE routers. To do this, include the interface statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit routing-instances routing-instance-name]  
  interface interface-name;
```

Specify both the physical and logical portions of the interface name, in the following format:

physical.logical

For example, in at-1/2/1.2, at-1/2/1 is the physical portion of the interface name and 2 is the logical portion. If you do not specify the logical portion of the interface name, 0 is set by default.

A logical interface can be associated with only one routing instance.



If you enable a routing protocol on all instances by specifying interfaces all when configuring the master instance of the protocol at the [edit protocols] hierarchy level and if you configure a specific interface for Layer 2 VPN routing at the [edit routing-instances *routing-instance-name*] hierarchy level, the latter interface statement takes precedence and the interface is used exclusively for the Layer 2 VPN.

If you explicitly configure the same interface name at both the [edit protocols] and [edit routing-instances *routing-instance-name*] hierarchy levels, when you try to commit the configuration, it will fail.

Configure CCC Encapsulation on Interfaces

You need to specify a circuit cross-connect (CCC) encapsulation type for each PE-router-to-CE-router interface running a Layer 2 VPN. This encapsulation type should match the encapsulation type configured under the routing instance. See “Configure the Encapsulation Type” on page 24 for information about how to configure the encapsulation type under the routing instance.

To configure the CCC encapsulation type, include the following statements at the [edit interfaces] hierarchy level:

```
[edit]  
  interfaces {  
    interface name {  
      encapsulation ccc-encapsulation-type;  
      unit unit number {  
        encapsulation ccc-encapsulation-type;  
      }  
    }  
  }
```

You configure the encapsulation type at the [edit interfaces] hierarchy level differently than you do at the [edit routing-instance] hierarchy level. For example, you specify the encapsulation as frame-relay at the [edit routing-instances] hierarchy level and as frame-relay-ccc at the [edit interfaces] hierarchy level.

You can run both standard Frame Relay and CCC Frame Relay on the same device. If you specify Frame Relay encapsulation (frame-relay-ccc) for the interface, you should also configure the encapsulation at the [edit interfaces *interface name* unit *unit-number*] hierarchy level as frame-relay-ccc. Otherwise, the logical interface unit defaults to standard Frame Relay.

For more information on how to configure interfaces and interface encapsulations, see the *JUNOS Internet Software Configuration Guide: Interfaces and Class of Service*.

Configure TCC Encapsulation on Interfaces

Also known as Layer 2.5 VPNs, the translation cross-connect (TCC) encapsulation types allow you to configure different encapsulation types at the ingress and egress of a Layer 2 VPN. For example, a CE router at the ingress of a Layer 2 VPN circuit can send traffic as Frame Relay. A CE router at the egress of that circuit can receive the traffic as ATM.

The configuration for TCC encapsulation types is similar to the configuration for CCC encapsulation types. Specify a TCC encapsulation type for each PE-router-to-CE-router interface running a Layer 2 VPN. This encapsulation type should match the encapsulation type configured under the routing instance. See “Configure the Encapsulation Type” on page 24 for information about how to configure the encapsulation type under the routing instance.

To configure the TCC encapsulation type, include the following statements at the [edit interfaces] hierarchy level:

```
[edit]
interfaces {
  interface name {
    encapsulation tcc-encapsulation-type;
    unit unit-number {
      encapsulation tcc-encapsulation-type;
    }
  }
}
```

You configure the encapsulation type at the [edit interfaces] hierarchy level differently from the [edit routing-instance] hierarchy level. For example, you specify the encapsulation as frame-relay at the [edit routing-instances] hierarchy level and as frame-relay-tcc at the [edit interfaces] hierarchy level.

For Layer 2.5 VPNs employing an Ethernet interface as the TCC router, you can configure an Ethernet translational cross-connect (TCC) or an extended VLAN TCC.

To configure an Ethernet TCC or an extended VLAN TCC, include the proxy and remote statements at the [edit interfaces *interface-name* unit *logical-unit-number* family tcc] hierarchy level:

```
[edit interfaces interfaces interface-name unit logical-unit-number family tcc]
proxy {
    inet-address address;
}
remote {
    (inet-address | mac-address) address;
}
```

The proxy inet-address address statement defines the IP address for which the TCC router is proxying.

The remote (inet-address | mac-address) statement defines the location of the remote router.

Ethernet TCC is supported on interfaces that carry IPv4 traffic only. Ethernet TCC encapsulation is supported on one-port Gigabit Ethernet, two-port Gigabit Ethernet, four-port Gigabit Ethernet, and four-port Fast Ethernet PICs only. Ethernet TCC is not supported on the T640 routing node.

For more information on how to configure interfaces and interface encapsulations, see the *JUNOS Internet Software Configuration Guide: Interfaces and Class of Service*.

Configure Layer 2 VPN Policing on Interfaces

You can use policing to control the amount of traffic flowing over the interfaces servicing a Layer 2 VPN. If policing is disabled on an interface, all the available bandwidth on a Layer 2 VPN tunnel can be used by a single CCC or TCC interface.

Layer 2 VPNs support only input policing. For more information about the policer statement, see the *JUNOS Internet Software Configuration Guide: Policy Framework*.

If you configure CCC encapsulation, then include the policer statement at the [edit interfaces *interface-name* unit *unit-number* family ccc] hierarchy level to enable Layer 2 VPN policing on an interface:

```
[edit]
interfaces interface-name {
    encapsulation encapsulation-type;
    unit 0 {
        family ccc {
            policer {
                input input-name;
            }
        }
    }
}
```

If you configure TCC encapsulation, then include the policer statement at the [edit interfaces *interface-name* unit *unit-number* family tcc] hierarchy level to enable Layer 2 VPN policing on an interface:

```
[edit]
interfaces interface-name {
  encapsulation encapsulation-type;
  unit 0 {
    family tcc {
      policer {
        input input-name;
      }
    }
  }
}
```

For information about how to configure the encapsulation type, see “Configure the Encapsulation Type” on page 24.

Configure the Route Distinguisher

Each routing instance that you configure on a PE router must have a unique route distinguisher associated with it. Both Layer 2 and Layer 3 VPNs need a route distinguisher to help Border Gateway Protocol (BGP) distinguish overlapping network layer reachability information (NLRIs) from different VPNs.

We recommend that you use unique route distinguishers for each routing instance that you configure. Although you can use the same route distinguisher on all PE routers in the same VPN, if you use a unique route distinguisher, you can determine the PE router from which a route originated.

To configure a route distinguisher on a PE router, include the route-distinguisher statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit routing-instances routing-instance-name]
route-distinguisher (as-number:number | ip-address:number);
```

The route distinguisher is a 6-byte value that you can specify in one of the following formats:

- *as-number:number*, where *as-number* is an autonomous system (AS) number (a 2-byte value) and *number* is any 4-byte value. The AS number can be in the range 1 through 65,535. We recommend that you use an Internet Assigned Numbers Authority (IANA)-assigned, nonprivate AS number, preferably the Internet service provider’s (ISP’s) own or the customer’s own AS number.
- *ip-address:number*, where *ip-address* is an IP address (a 4-byte value) and *number* is any 2-byte value. The IP address can be any globally unique unicast address. We recommend that you use the address that you configure in the router-id statement, which is a nonprivate address in your assigned prefix range.

If you configure the route-distinguisher-id statement at the [edit routing-options] hierarchy level, a route distinguisher is automatically assigned to the routing instance. If you configure the route-distinguisher statement in addition to the route-distinguisher-id statement, the value configured for route-distinguisher supersedes the value generated from route-distinguisher-id.

The route-distinguisher-id statement is configured as follows:

```
[edit]
routing-options {
  route-distinguisher-id ip-address;
}
```

When configured, a type 1 route distinguisher is automatically assigned to the routing instance using the format *ip-address:number*. The IP address is specified by the route-distinguisher-id statement and the number is unique for the routing instance.

Disable the Control Word for Layer 2 VPNs

The emulated VC encapsulation for Layer 2 VPNs is accomplished by adding a 4-byte control word between the Layer 2 protocol data unit (PDU) being transported and the VC label that is used for demultiplexing. Various networking formats (ATM, Frame Relay, Ethernet, and so on) use the control word in a variety of ways.

JUNOS software does not support the control word for any networking format, meaning that it is not fully compliant with the Internet draft in cases where the control word is mandatory. To be minimally compliant with the Internet draft, JUNOS supports a null control word (a control word of all zeros). This null control word is configured by default. If JUNOS receives a packet with a control word attached, the control word is discarded before the packet is forwarded to its destination.

If you have configured Layer 2 VPNs on a network where some routers are running JUNOS 5.6 and some routers are running earlier JUNOS releases, you need to include the no-control-word statement at the [edit routing-instances *routing-instance-name* protocols l2vpn] hierarchy level on the JUNOS 5.6 routers:

```
[edit routing-instances routing-instance-name protocols l2vpn]
no-control-word;
```

This is not necessary when configuring Layer 2 circuits. For more information, see “Disable the Control Word for Layer 2 Circuits” on page 329.

Configure Policy for the Layer 2 VPN Routing Table

You can configure import and export policies for the Layer 2 VPN routing table. The procedure is identical to the one used for the Layer 3 VPN routing table. For more information, see “Configure Policy for the PE Router’s VRF Table” on page 89.

You can also use the vpn-apply-export statement to apply both VPN and BGP export policies to the Layer 2 VPN routing table. For more information, see “Apply Both the VRF Export and the BGP Export Policies” on page 92.

Configure the Connections to the Local Site

For each local site, the PE router advertises a set of VPN labels to the other PE routers servicing the Layer 2 VPN. The VPN labels constitute a single block of contiguous labels; however, to allow for reprovisioning, more than one such block can be advertised. Each label block consists of a label base, a range (the size of the block), and a remote site ID that identifies the sequence of remote sites that connect to the local site using this label block (the remote site ID is the first site identifier in the sequence). The encapsulation type is also advertised along with the label block.

To configure the connections to the local site on the PE router, perform the following tasks:

- Configure the Site on page 23
- Configure the Remote Site ID on page 24
- Configure the Encapsulation Type on page 24
- Trace Layer 2 VPN Traffic and Operations on page 25

Configure the Site

All the Layer 2 circuits provisioned for a local site are listed as the set of logical interfaces (using the interface statement) within the site statement.

On each PE router, you must configure each site that has a circuit to the PE router. To do this, include the site statement at the [edit routing-instances *routing-instance-name* protocols l2vpn] hierarchy level:

```
[edit routing-instances routing-instance-name protocols l2vpn]
site site-name {
  site-identifier identifier;
  interface interface-name {
    remote-site-id remote-site-ID;
  }
}
```

You must configure the following for each site:

- site—Name of the site.
- site-identifier—Unsigned 16-bit number greater than zero that uniquely identifies the site.
- interface—A name for the interface and, optionally, a remote site ID for remote site connections. See “Configure the Remote Site ID” on page 24.

Configure the Remote Site ID

The remote site ID allows you to configure a sparse Layer 2 VPN topology. This means that each site does not have to connect to all the other sites in the VPN, making it unnecessary to allocate circuits for all the remote sites. Remote site IDs are particularly important if you configure a topology more complicated than full-mesh, such as a hub-and-spoke topology.

The remote site ID (configured with the `remote-site-id` statement) corresponds to the site ID (configured with the `site-identifier` statement) configured at a separate site.

For example, a configuration for Router PE1 connected to Router CE1 is as follows:

```
site-identifier 1;
interface so-0/0/0 {
    remote-site-id 2;
}
```

The configuration for Router PE2 connected to Router CE2 is then as follows:

```
site-identifier 2;
interface so-0/0/1 {
    remote-site-id 1;
}
```

Configure the `remote-site-id` statement at the [edit routing-instances *routing-instance-name* protocols l2vpn site *site-name* interface *interface-name*] hierarchy level:

```
[edit routing-instances routing-instance-name protocols l2vpn]
site site-name {
    interface interface-name {
        remote-site-id remote-site-ID;
    }
}
```

If you do not explicitly configure the `remote-site-id` statement for the interface configured at the [edit routing-instances *routing-instance-name* protocols l2vpn site *site-name*] hierarchy level, a remote site ID is assigned to that interface.

The remote site ID for an interface is automatically set to 1 higher than the remote site ID for the previous interface. The order of the interfaces is based on their `site-identifier` statements. For example, if the first interface in the list does not have a remote site ID, its ID is set to 1. The second interface in the list has its remote site ID set to 2, and the third has its remote site ID set to 3. The remote site IDs of any interfaces that follow are incremented in the same manner if you do not explicitly configure them.

Configure the Encapsulation Type

The encapsulation type you configure at each Layer 2 VPN site varies depending on which Layer 2 protocol you choose to configure. You need to use the same protocol at each Layer 2 VPN site if you configure ethernet-vlan as the encapsulation type. You do *not* need to use the same protocol at each Layer 2 VPN site if you configure any of the following encapsulation types:

- atm-aal5—ATM Adaptation Layer (AAL/5)
- atm-cell—ATM cell

- cisco-hdlc—Cisco Systems-compatible High-level Data Link Control (HDLC)
- ethernet—Ethernet
- ethernet-vlan—Ethernet VLAN
- frame-relay—Frame Relay
- interworking—Layer 2.5 interworking VPN
- ppp—Point-to-Point Protocol (PPP)

If you configure different protocols at your Layer 2 VPN sites, you need to configure a TCC encapsulation type. For more information, see “Configure TCC Encapsulation on Interfaces” on page 19.

To configure the Layer 2 protocol accepted by the PE router, specify the encapsulation type by including the encapsulation statement at the [edit routing-instances *routing-instance-name* protocols l2vpn] hierarchy level:

```
[edit routing-instances routing-instance-name protocols l2vpn]
encapsulation type
```

Trace Layer 2 VPN Traffic and Operations

To trace Layer 2 VPN protocol traffic, you can specify options in the Layer 2 VPN traceoptions statement at the [edit routing-instances *routing-instance-name* protocols l2vpn] hierarchy level:

```
[edit routing-instances routing-instance-name protocols l2vpn]
traceoptions {
  file filename <replace> <size size> <files number> <nostamp>;
  flag flag <flag-modifier> <disable>;
}
```

The following trace flags display the operations associated with Layer 2 VPNs. Each can carry one or more of the following modifiers:

- all—All Layer 2 VPN tracing options
- connections—Layer 2 VPN connections (events and state changes)
- error—Error conditions
- nlri—Layer 2 VPN advertisements received or sent using BGP
- route—Trace routing information
- topology—Layer 2 VPN topology changes due to reconfiguration or due to advertisements received from other PE routers using BGP

Disable Normal TTL Decrementing for VPNs

To diagnose networking problems related to VPNs (Layer 2 or Layer 3), it can be useful to disable normal time-to-live (TTL) decrementing. In JUNOS, you can do this with the `no-propagate-ttl` and `no-decrement-ttl` statements. However, when tracing VPN traffic, only the `no-propagate-ttl` statement is effective.

For the `no-propagate-ttl` statement to have an effect on VPN behavior, you need to clear the PE-router-to-PE-router BGP session, or disable and then enable the VPN routing instance.

For more information about the `no-propagate-ttl` and `no-decrement-ttl` statements, see the *JUNOS Internet Software Configuration Guide: MPLS Applications*.

Configure Packet Forwarding for VPNs

You can configure the router to support packet forwarding for Internet Protocol version 4 (IPv4) traffic in Layer 2 and Layer 3 VPNs. Packet forwarding is handled in one of the following ways, depending on the type of helper service configured:

- **BOOTP service**—Clients send Bootstrap Protocol (BOOTP) requests through the router configured with BOOTP service to a server in the specified routing instance. The server recognizes the client address and sends a response back to the router configured with BOOTP service. This router forwards the reply to the correct client address in the specified routing instance.
- **Other services**—Clients send requests through the router configured with the service to a server in the specified routing instance. The server recognizes the client address and sends a response to the correct client address in the specified routing instance.

To enable packet forwarding for VPNs, configure the `helpers` statement at the `[edit forwarding-options]` hierarchy level as follows:

```
[edit forwarding-options]
helpers {
  service {
    description description-of-service;
    server {
      address address {
        routing-instance routing-instance-names;
      }
    }
  }
  interface interface-name {
    description description-of-interface;
    no-listen;
    server {
      address address {
        routing-instance routing-instance-names;
      }
    }
  }
}
```



Note

You can enable packet forwarding for multiple VPNs. However, the client and server must be within the same VPN. Any Juniper Networks routers with packet forwarding enabled along the path between the client and server must also reside within the same VPN.

The address and routing instance together constitute a unique server. This has implications for routers configured with BOOTP service, which can accept multiple servers.

For example, a BOOTP service could be configured as follows:

```
[edit forwarding-options helpers bootp]
{
  server address 1.2.3.4 routing-instance [instance-A instance-B];
}
```

Though the addresses are identical, the routing instances are different. A packet coming in for BOOTP service on instance-A is forwarded to 1.2.3.4 in the instance-A routing instance, while a packet coming in on instance-B is forwarded in the instance-B routing instance. Other services can only accept a single server, so this configuration does not apply in those cases.

For more information about the statements configured at the [edit forwarding-options] hierarchy level, see the *JUNOS Internet Software Configuration Guide: Policy Framework*.

Configure VPN Graceful Restart

For information on how to configure VPN graceful restart, see “Configure VPN Graceful Restart” on page 105.

.....

Chapter 4

Layer 2 VPN Configuration Example

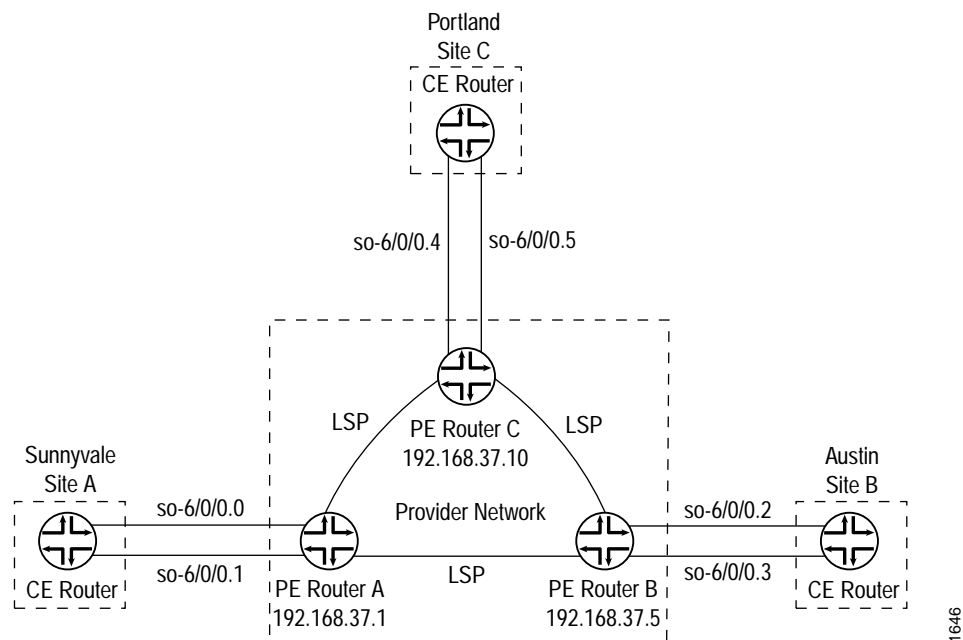
This chapter provides an example of a Layer 2 virtual private network (VPN) spanning three sites. The following sections explain how to configure Layer 2 VPN functionality on the provider edge (PE) routers connected to each site:

- Simple Full-Mesh Layer 2 VPN Overview on page 29
- Enable an IGP on the PE Routers on page 30
- Configure MPLS LSP Tunnels between the PE Routers on page 31
- Configure IBGP on the PE Routers on page 32
- Configure Routing Instances for Layer 2 VPNs on the PE Routers on page 34
- Configure CCC Encapsulation on the Interfaces on page 36
- Configure VPN Policy on the PE Routers on page 37
- Layer 2 VPN Configuration Summarized by Router on page 39

Simple Full-Mesh Layer 2 VPN Overview

In the sections that follow, you configure a simple full-mesh Layer 2 VPN spanning three sites: Sunnyvale, Austin, and Portland. Each site connects to a PE router. The customer edge (CE) routers at each site use Frame Relay to carry Layer 2 traffic to the PE routers. Since this example uses a full-mesh topology between all three sites, each site requires two logical interfaces (one for each of the other CE routers), although only one physical link is needed to connect each PE router to each CE router. Figure 3 illustrates the topology of this Layer 2 VPN.

Figure 3: Example of a Simple Full-Mesh Layer 2 VPN Topology



Enable an IGP on the PE Routers

To allow the PE routers to exchange routing information among themselves, you must configure an interior gateway protocol (IGP) or static routes on these routers. You configure the IGP on the master instance of the routing protocol process (`rp`) (that is, at the `[edit protocols]` hierarchy level), not within the Layer 2 VPN routing instance (that is, not at the `[edit routing-instances]` hierarchy level). Turn on traffic engineering on the IGP.

You configure the IGP in the standard way. This example does not include this portion of the configuration.

Configure MPLS LSP Tunnels between the PE Routers

In this configuration example, Resource Reservation Protocol (RSVP) is used for Multiprotocol Label Switching (MPLS) signaling. Therefore, in addition to configuring RSVP, you must create an MPLS label-switched path (LSP) to tunnel the VPN traffic.

On Router A, enable RSVP and configure one end of the MPLS LSP tunnel to Router B. When configuring the MPLS LSP, include all interfaces using the interface all statement.

```
[edit]
protocols {
  rsvp {
    interface all;
  }
  mpls {
    label-switched-path RouterA-to-RouterB {
      to 192.168.37.5;
      primary Path-to-RouterB;
    }
    label-switched-path RouterA-to-RouterC {
      to 192.168.37.10;
      primary Path-to-RouterC;
    }
  }
  interface all;
}
```

On Router B, enable RSVP and configure the other end of the MPLS LSP tunnel. Again, configure the interfaces using the interface all statement.

```
[edit]
protocols {
  rsvp {
    interface all;
  }
  mpls {
    label-switched-path RouterB-to-RouterA {
      to 192.168.37.1;
      primary Path-to-RouterA;
    }
    label-switched-path RouterB-to-RouterC {
      to 192.168.37.10;
      primary Path-to-RouterC;
    }
  }
  interface all;
}
```

On Router C, enable RSVP and configure the other end of the MPLS LSP tunnel. Again, configure all interfaces using the interface all statement.

```
[edit]
protocols {
  rsvp {
    interface all;
  }
  mpls {
    label-switched-path RouterC-to-RouterA {
      to 192.168.37.1;
      primary Path-to-RouterA;
    }
    label-switched-path RouterC-to-RouterB {
      to 192.168.37.5;
      primary Path-to-RouterB;
    }
    interface all;
  }
}
```

Configure IBGP on the PE Routers

On the PE routers, configure an internal Border Gateway Protocol (IBGP) session with the following parameters:

- **Layer 2 VPN**—To indicate that the IBGP session is for a Layer 2 VPN, include the family l2vpn statement.
- **Local address**—The IP address in the local-address statement is the same as the address configured in the to statement at the [edit protocols mpls label-switched-path *lsp-path-name*] hierarchy level. The IBGP session for Layer 2 VPNs runs through this address.
- **Neighbor address**—Include the neighbor statement, specifying the IP address of the neighboring PE router.

On Router A, configure IBGP as follows:

```
[edit]
protocols{
  bgp {
    import match-all;
    export match-all;
    group pe-pe {
      type internal;
      neighbor 192.168.37.5 {
        local-address 192.168.37.1;
        family l2vpn {
          unicast;
        }
      }
    }
  }
}
```



```

neighbor 192.168.37.10 {
  local-address 192.168.37.1;
  family l2vpn {
    unicast;
  }
}
}
}

```

On Router B, configure IBGP as follows:

```

[edit]
protocols{
  bgp {
    local-address 192.168.37.5;
    import match-all;
    export match-all;
    group pe-pe {
      type internal;
      neighbor 192.168.37.1 {
        local-address 192.168.37.5;
        family l2vpn {
          unicast;
        }
      }
      neighbor 192.168.37.10 {
        local-address 192.168.37.5;
        family l2vpn {
          unicast;
        }
      }
    }
  }
}
}

```

On Router C, configure IBGP as follows:

```

[edit]
protocols{
  bgp {
    local-address 192.168.37.10;
    import match-all;
    export match-all;
    group pe-pe {
      type internal;
      neighbor 192.168.37.1 {
        local-address 192.168.37.10;
        family l2vpn {
          unicast;
        }
      }
      neighbor 192.168.37.5 {
        local-address 192.168.37.10;
        family l2vpn {
          unicast;
        }
      }
    }
  }
}
}

```

Configure Routing Instances for Layer 2 VPNs on the PE Routers

The three PE routers service the Layer 2 VPN, so you need to configure a routing instance on each router. For the VPN, you must define the following in each routing instance:

- Route distinguisher, which must be unique for each routing instance on the PE router. It is used to distinguish the addresses in one VPN from those in another VPN.
- Instance type of l2vpn, which configures the router to run a Layer 2 VPN.
- Interfaces connected to the CE routers.
- VPN routing and forwarding (VRF) import and export policies, which must be the same on each PE router that services the same VPN and are used to control the network topology. Unless the import policy contains only a then reject statement, it must include a reference to a community. Otherwise, when you attempt to commit the configuration, the commit fails.

On Router A, configure the following routing instances for the Layer 2 VPN:

```
[edit]
routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    instance-type l2vpn;
    interface so-6/0/0.0;
    interface so-6/0/0.1;
    route-distinguisher 100:1;
    vrf-import vpn-SPA-import;
    vrf-export vpn-SPA-export;
    protocols {
      l2vpn {
        encapsulation-type frame-relay;
        site Sunnyvale {
          site-identifier 1;
          interface so-6/0/0.0 {
            remote-site-id 2;
          }
          interface so-6/0/0.1 {
            remote-site-id 3;
          }
        }
      }
    }
  }
}
```

On Router B, configure the following routing instance:

```
[edit]
routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    instance-type l2vpn;
    interface so-6/0/0.2;
    interface so-6/0/0.3;
    route-distinguisher 100:1;
    vrf-import vpn-SPA-import;
    vrf-export vpn-SPA-export;
```

```

protocols {
  l2vpn {
    encapsulation-type frame-relay;
    site Austin {
      site-identifier 2;
      interface so-6/0/0.2 {
        remote-site-id 1;
      }
      interface so-6/0/0.3 {
        remote-site-id 3;
      }
    }
  }
}

```

On Router C, configure the following routing instance for the Layer 2 VPN:

```

[edit]
routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    instance-type l2vpn;
    interface so-6/0/0.4;
    interface so-6/0/0.5;
    route-distinguisher 100:1;
    vrf-import vpn-SPA-import;
    vrf-export vpn-SPA-export;
    protocols {
      l2vpn {
        encapsulation-type frame-relay;
        site Portland {
          site-identifier 3;
          interface so-6/0/0.4 {
            remote-site-id 1;
          }
          interface so-6/0/0.5 {
            remote-site-id 2;
          }
        }
      }
    }
  }
}

```

Configure CCC Encapsulation on the Interfaces

You need to specify a circuit cross-connect (CCC) encapsulation type for each PE-router-to-CE-router interface running in the Layer 2 VPN. This encapsulation type should match the encapsulation type configured under the routing instance.

Configure the following CCC encapsulation types for the interfaces on Router A:

```
[edit]
interfaces {
  interface so-6/0/0.0 {
    encapsulation frame-relay-ccc;
    unit 0 {
      encapsulation frame-relay-ccc;
    }
  }
  interface so-6/0/0.1 {
    encapsulation frame-relay-ccc;
    unit 1 {
      encapsulation frame-relay-ccc;
    }
  }
}
```

Configure the following CCC encapsulation types for the interfaces on Router B:

```
[edit]
interfaces {
  interface so-6/0/0.2 {
    encapsulation frame-relay-ccc;
    unit 2 {
      encapsulation frame-relay-ccc;
    }
  }
  interface so-6/0/0.3 {
    encapsulation frame-relay-ccc;
    unit 3 {
      encapsulation frame-relay-ccc;
    }
  }
}
```

Configure the following CCC encapsulation types for the interfaces on Router C:

```
[edit]
interfaces {
  interface so-6/0/0.4 {
    encapsulation frame-relay-ccc;
    unit 4 {
      encapsulation frame-relay-ccc;
    }
  }
  interface so-6/0/0.5 {
    encapsulation frame-relay-ccc;
    unit 5 {
      encapsulation frame-relay-ccc;
    }
  }
}
```

Configure VPN Policy on the PE Routers

You must configure VPN import and export policies on each of the PE routers so that they install the appropriate routes in their VRF tables, which the routers use to forward packets within the VPN.

On Router A, configure the following VPN import and export policies:

```
[edit]
policy-options {
  policy-statement match-all {
    term acceptable {
      then accept;
    }
  }
  policy-statement vpn-SPA-export {
    term a {
      then {
        community add SPA-com;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  policy-statement vpn-SPA-import {
    term a {
      from {
        protocol bgp;
        community SPA-com;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  community SPA-com members target:69:100;
}
```

On Router B, configure the following VPN import and export policies:

```
[edit]
policy-options {
  policy-statement match-all {
    term acceptable {
      then accept;
    }
  }
  policy-statement vpn-SPA-import {
    term a {
      from {
        protocol bgp;
        community SPA-com;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
}
```

```

policy-statement vpn-SPA-export {
  term a {
    then {
      community add SPA-com;
      accept;
    }
  }
  term b {
    then reject;
  }
}
community SPA-com members target:69:100;
}

```

On Router C, configure the following VPN import and export policies:

```

[edit]
policy-options {
  policy-statement match-all {
    term acceptable {
      then accept;
    }
  }
  policy-statement vpn-SPA-import {
    term a {
      from {
        protocol bgp;
        community SPA-com;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement vpn-SPA-export {
    term a {
      then {
        community add SPA-com;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community SPA-com members target:69:100;
}

```

To apply the VPN policies on the routers, include the `vrf-export` and `vrf-import` statements when you configure the routing instance. The VRF import and export policies handle the route distribution across the IBGP session running between the PE routers.

To apply the VPN policies on Router A, include the following statements:

```

[edit]
routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    vrf-import vpn-SPA-import;
    vrf-export vpn-SPA-export;
  }
}

```

To apply the VPN policies on Router B, include the following statements:

```
[edit]
routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    vrf-import vpn-SPA-import;
    vrf-export vpn-SPA-export;
  }
}
```

To apply the VPN policies on Router C, include the following statements:

```
[edit]
routing-instances {
  VPN-Sunnyvale-Portland-Austin {
    vrf-import vpn-SPA-import;
    vrf-export vpn-SPA-export;
  }
}
```

Layer 2 VPN Configuration Summarized by Router

Summary for Router A (PE Router for Sunnyvale)

Routing Instance for VPN	<pre>routing-instances { VPN-Sunnyvale-Portland-Austin{ instance-type l2vpn; interface so-6/0/0.0; interface so-6/0/0.1; route-distinguisher 100:1; vrf-import vpn-SPA-import; vrf-export vpn-SPA-export; } }</pre>
Configure Layer 2 VPN	<pre>protocols { l2vpn { encapsulation-type frame-relay; site Sunnyvale { site-identifier 1; interface so-6/0/0.0 { remote-site-id 2; } interface so-6/0/0.1 { remote-site-id 3; } } } }</pre>

```

Configure CCC Encapsulation Types for Interfaces
interfaces {
  interface so-6/0/0.0 {
    encapsulation frame-relay-ccc;
    unit 0 {
      encapsulation frame-relay-ccc;
    }
  }
  interface so-6/0/0.1 {
    encapsulation frame-relay-ccc;
    unit 1 {
      encapsulation frame-relay-ccc;
    }
  }
}

Master Protocol Instance
protocols {

  Enable RSVP
  rsvp {
    interface all;
  }

  Configure MPLS LSPs
  mpls {
    label-switched-path RouterA-to-RouterB {
      to 192.168.37.5;
      primary Path-to-RouterB {
        cspf;
      }
    }
    label-switched-path RouterA-to-RouterC {
      to 192.168.37.10;
      primary Path-to-RouterC {
        cspf;
      }
    }
    interface all;
  }

  Configure IBGP
  bgp {
    import match-all;
    export match-all;
    group pe-pe {
      type internal;
      neighbor 192.168.37.5 {
        local-address 192.168.37.1;
        family l2vpn {
          unicast;
        }
      }
      neighbor 192.168.37.10 {
        local-address 192.168.37.1;
        family l2vpn {
          unicast;
        }
      }
    }
  }
}

```



```

Configure VPN Policy policy-options {
    policy-statement match-all {
        term acceptable {
            then accept;
        }
    }
    policy-statement vpn-SPA-export {
        term a {
            then {
                community add SPA-com;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    policy-statement vpn-SPA-import {
        term a {
            from {
                protocol bgp;
                community SPA-com;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    community SPA-com members target:69:100;
}

```

Summary for Router B (PE Router for Austin)

```

Routing Instance for VPN routing-instances {
    VPN-Sunnyvale-Portland-Austin {
        instance-type l2vpn;
        interface so-6/0/0.2;
        interface so-6/0/0.3;
        route-distinguisher 100:1;
        vrf-import vpn-SPA-import;
        vrf-export vpn-SPA-export;
    }
}

Configure Layer 2 VPN protocols {
    l2vpn {
        encapsulation-type frame-relay;
        site Austin {
            site-identifier 2;
            interface so-6/0/0.2 {
                remote-site-id 1;
            }
            interface so-6/0/0.3 {
                remote-site-id 3;
            }
        }
    }
}

```

```

[edit]
Configure CCC Encapsulation Types for Interfaces
    interfaces {
        interface so-6/0/0.2 {
            encapsulation frame-relay-ccc;
            unit 2 {
                encapsulation frame-relay-ccc;
            }
        }
        interface so-6/0/0.3 {
            encapsulation frame-relay-ccc;
            unit 3 {
                encapsulation frame-relay-ccc;
            }
        }
    }

Master Protocol Instance protocols {

    Enable RSVP
        rsvp {
            interface all;
        }

    Configure MPLS LSPs
        mpls {
            label-switched-path RouterB-to-RouterA {
                to 192.168.37.1;
                primary Path-to-RouterA{
                    cspf;
                }
            }
            label-switched-path RouterB-to-RouterC {
                to 192.168.37.10;
                primary Path-to-RouterC {
                    cspf;
                }
            }
            interface all;
        }

    Configure IBGP
        bgp {
            local-address 192.168.37.5;
            import match-all;
            export match-all;
            group pe-pe {
                type internal;
                neighbor 192.168.37.1 {
                    local-address 192.168.37.5;
                    family l2vpn {
                        unicast;
                    }
                }
                neighbor 192.168.37.10 {
                    local-address 192.168.37.5;
                    family l2vpn {
                        unicast;
                    }
                }
            }
        }
    }

```

```

Configure VPN Policy policy-options {
    policy-statement match-all {
        term acceptable {
            then accept;
        }
    }
    policy-statement vpn-SPA-import {
        term a {
            from {
                protocol bgp;
                community SPA-com;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement vpn-SPA-export {
        term a {
            then {
                community add SPA-com;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    community SPA-com members target:69:100;
}

```

Summary for Router C (PE Router for Portland)

```

Routing Instance for VPN routing-instances {
    VPN-Sunnyvale-Portland-Austin {
        instance-type l2vpn;
        interface so-6/0/0.3;
        interface so-6/0/0.4;
        route-distinguisher 100:1;
        vrf-import vpn-SPA-import;
        vrf-export vpn-SPA-export;
    }
}

Configure Layer 2 VPN protocols {
    l2vpn {
        encapsulation-type frame-relay;
        site Portland {
            site-identifier 3;
            interface so-6/0/0.4 {
                remote-site-id 1;
            }
            interface so-6/0/0.5 {
                remote-site-id 2;
            }
        }
    }
}
}

```

```

[edit]
Configure CCC Encapsulation Types for Interfaces
    interfaces {
        interface so-6/0/0.4 {
            encapsulation frame-relay-ccc;
            unit 4 {
                encapsulation frame-relay-ccc;
            }
        }
        interface so-6/0/0.5 {
            encapsulation frame-relay-ccc;
            unit 5 {
                encapsulation frame-relay-ccc;
            }
        }
    }
}

Master Protocol Instance protocols {

    Enable RSVP
        rsvp {
            interface all;
        }

    Configure MPLS LSPs
        mpls {
            label-switched-path RouterC-to-RouterA {
                to 192.168.37.1;
                primary Path-to-RouterA {
                    cspf;
                }
            }
            label-switched-path RouterC-to-RouterB {
                to 192.168.37.5;
                primary Path-to-RouterB {
                    cspf;
                }
            }
            interface all;
        }

    Configure IBGP
        bgp {
            local-address 192.168.37.10;
            import match-all;
            export match-all;
            group pe-pe {
                type internal;
                neighbor 192.168.37.1 {
                    local-address 192.168.37.10;
                    family l2vpn {
                        unicast;
                    }
                }
                neighbor 192.168.37.5 {
                    local-address 192.168.37.10;
                    family l2vpn {
                        unicast;
                    }
                }
            }
        }
    }
}

```

```

Configure VPN Policy policy-options {
    policy-statement match-all {
        term acceptable {
            then accept;
        }
    }
    policy-statement vpn-SPA-import {
        term a {
            from {
                protocol bgp;
                community SPA-com;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement vpn-SPA-export {
        term a {
            then {
                community add SPA-com;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    community SPA-com members target:69:100;
}

```


Chapter 5

Summary of Layer 2 VPN Configuration Statements

The following sections explain the major routing-instances configuration statements that apply specifically to Layer 2 virtual private networks (VPNs). The statements are organized alphabetically. Routing instances and the statements at the [edit routing-instances *routing-instance-name* protocols] hierarchy level are explained in the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

control-word

Syntax	(control-word no-control-word);
Hierarchy Level	[edit protocols l2circuit neighbor <i>address</i> interface <i>interface-name</i>] [edit routing-instances <i>routing-instance-name</i> protocols l2vpn]
Description	<p>The control word is 4 bytes long and is inserted between the Layer 2 protocol data unit (PDU) being transported and the VC label that is used for demultiplexing.</p> <ul style="list-style-type: none">■ control-word—Enables the use of the control word. Default: The control word is enabled by default. You can also configure the control word explicitly using the control-word statement.■ no-control-word—Disables the use of the control word.
Usage Guidelines	See “Disable the Control Word for Layer 2 VPNs” on page 22.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

description

Syntax	description <i>text</i> ;
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i>]
Description	<p>Allows you to provide a textual description of the routing instance. Enclose any descriptive text that includes spaces in quotation marks (" "). Any descriptive text you include is displayed in the output of the show route instance detail command and has no effect on the operation of the routing instance.</p>
Usage Guidelines	See “Configure the Description” on page 17.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

encapsulation

You configure three encapsulation types: one for the encapsulation on the physical interfaces, one for the encapsulation on the logical interface, and one for the Layer 2 protocol on the routing instance.

encapsulation (physical interface)

Syntax `encapsulation (atm-ccc-cell-relay | atm-pvc | cisco-hdlc | cisco-hdlc-ccc | cisco-hdlc-tcc | ethernet-ccc | frame-relay | frame-relay-ccc | frame-relay-tcc | multilink-frame-relay-uni-nni | ppp | ppp-ccc | ppp-tcc | vlan-ccc | extended-vlan-ccc);`

Hierarchy Level [edit interfaces *interface-name*]

Description Physical link-layer encapsulation type.

Options atm-ccc-cell-relay—Use Asynchronous Transfer Mode (ATM) cell relay encapsulation.

atm-pvc—Use ATM permanent virtual connection (PVC) encapsulation.

cisco-hdlc—Use Cisco-compatible High-level Data Link Control (HDLC) framing.

cisco-hdlc-ccc—Use Cisco-compatible HDLC framing on circuit cross-connect (CCC) circuits.

cisco-hdlc-tcc—Use Cisco-compatible HDLC framing on translational cross-connect (TCC) circuits for connecting unlike media.

ethernet-ccc—Use Ethernet CCC encapsulation on Ethernet interfaces that must accept packets carrying standard Tag Protocol ID (TPID) values.

ethernet-tcc—For interfaces that carry Internet Protocol Version 4 (IPv4) traffic, use Ethernet TCC encapsulation on interfaces that must accept packets carrying standard TPID values. Ethernet TCC is not currently supported on Fast Ethernet 48-port Physical Interface Cards (PICs) or the T-series platforms.

extended-vlan-ccc—Use extended virtual local area network (VLAN) encapsulation on CCC circuits with Gigabit Ethernet and four-port Fast Ethernet interfaces that must accept packets carrying 802.1Q values.

extended-vlan-tcc—For interfaces that carry IPv4 traffic, use extended VLAN encapsulation on TCC circuits with Gigabit Ethernet interfaces on which you want to use 802.1Q tagging. Extended Ethernet TCC is not currently supported on Fast Ethernet 48-port PICs or the T-series platforms.

frame-relay—Use Frame Relay encapsulation.

frame-relay-ccc—Use plain Frame Relay encapsulation or Frame Relay encapsulation on CCC circuits.

frame-relay-tcc—Use Frame Relay encapsulation on TCC circuits for connecting unlike media.

multilink-frame-relay-uni-nni—Use only on link services interfaces functioning as FRF.16 bundles and their constituent T1, E1, or NxDS-0 interfaces.

ppp—Use serial Point-to-Point Protocol (PPP) encapsulation.

	ppp-ccc—Use serial PPP encapsulation on CCC circuits. When you use this encapsulation, you can configure the family ccc only.
	ppp-tcc—Use serial PPP encapsulation on TCC circuits for connecting unlike media. When you use this encapsulation, you can configure the family tcc only.
	vlan-ccc—Use Ethernet VLAN encapsulation on CCC circuits.
Default	PPP encapsulation.
Usage Guidelines	See “Configure CCC Encapsulation on Interfaces” on page 18 or “Configure TCC Encapsulation on Interfaces” on page 19.
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.

encapsulation (logical interface)

Syntax	encapsulation (atm-ccc-cell-relay atm-ccc-vc-mux atm-tcc-vc-mux atm-cisco-nlpid atm-nlpid atm-snap atm-tcc-snap atm-vc-mux ether-over-atm-llc frame-relay-ccc frame-relay-tcc multilink-frame-relay-end-to-end multilink-ppp vlan-ccc);
Hierarchy Level	[edit interfaces <i>interface-name</i> unit <i>logical-unit-number</i>]
Description	Logical link-layer encapsulation type.
Options	<p>atm-ccc-cell-relay—Use ATM cell relay encapsulation.</p> <p>atm-ccc-vc-mux—Use ATM virtual connection (VC) multiplex encapsulation on CCC circuits. When you use this encapsulation, you can configure the family ccc only.</p> <p>atm-tcc-vc-mux—Use ATM VC multiplex encapsulation on TCC circuits. When you use this encapsulation, you can configure the family tcc only.</p> <p>atm-cisco-nlpid—Use Cisco ATM NLPID encapsulation. When you use this encapsulation, you can configure the family inet only.</p> <p>atm-nlpid—Use ATM NLPID encapsulation. When you use this encapsulation, you can configure the family inet only.</p> <p>atm-snap—Use ATM Subnetwork Access Protocol (SNAP) encapsulation.</p> <p>atm-tcc-snap—Use ATM SNAP encapsulation on TCC circuits.</p> <p>atm-vc-mux—Use ATM VC multiplex encapsulation. When you use this encapsulation, you can configure the family inet only.</p> <p>ether-over-atm-llc—For interfaces that carry IPv4 traffic, use Ethernet over ATM Logical Link Control (LLC) encapsulation. When you use this encapsulation, you cannot configure multipoint interfaces.</p> <p>frame-relay-ccc—Use Frame Relay encapsulation on CCC circuits. When you use this encapsulation, you can configure the family ccc only.</p> <p>frame-relay-tcc—Use Frame Relay encapsulation on TCC circuits for connecting unlike media. When you use this encapsulation, you can configure the family tcc only.</p>

multilink-frame-relay-end-to-end—Use Multilink Frame Relay (MLFR) FRF.15 encapsulation for multilink and link services interfaces and their constituent T1 or E1 interfaces.

multilink-ppp—Use Multilink Point-to-Point Protocol (MLPPP) encapsulation. This encapsulation is used only on multilink interfaces and their constituent T1 or E1 interfaces.

vlan-ccc—Use Ethernet VLAN encapsulation on CCC circuits. When you use this encapsulation, you can configure the family ccc only.

Usage Guidelines See “Configure CCC Encapsulation on Interfaces” on page 18 or “Configure TCC Encapsulation on Interfaces” on page 19.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

encapsulation (Layer 2 VPN)

Syntax `encapsulation type`

Hierarchy Level [edit routing-instances *routing-instance-name* protocols l2vpn]

Description Layer 2 protocol used for traffic from the customer edge (CE) router.

Options *type*—The following Layer 2 encapsulation types are supported:

- atm-aal5—ATM Adaptation Layer (AAL/5)
- atm-cell—ATM cell
- cisco-hdlc—Cisco Systems-compatible HDLC
- ethernet—Ethernet
- ethernet-vlan—Ethernet VLAN
- frame-relay—Frame Relay
- interworking—Layer 2.5 interworking VPN
- ppp—PPP

Usage Guidelines See “Configure the Encapsulation Type” on page 24.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

See also encapsulation on page 48

instance-type

Syntax	instance-type l2vpn;
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i>]
Description	Type of routing instance.
Options	l2vpn—Layer 2 VPN instance.
Usage Guidelines	See “Configure the Instance Type” on page 17.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

interface

Syntax	interface <i>interface-name</i> ;
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i>] [edit routing-instances <i>routing-instance-name</i> protocols l2vpn site <i>site-name</i>]
Description	Interface over which Layer 2 VPN traffic travels between the provider edge (PE) router and the CE router. You configure the interface on the PE router. If the instance type is l2vpn, the interface statement is required.
Options	<i>interface-name</i> —Name of the interface to configure.
Usage Guidelines	See “Configure Interfaces for Layer 2 VPN Routing” on page 18.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

l2vpn

Syntax	l2vpn {...}
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i> instance-type] [edit routing-instances <i>routing-instance-name</i> protocols]
Description	Enable a Layer 2 VPN on the routing instance.
Usage Guidelines	See “Configure Routing Instances for Layer 2 VPNs on the PE Routers” on page 16.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

no-control-word

See “control-word” on page 47.

proxy

Syntax	proxy inet-address <i>address</i> ;
Hierarchy Level	[edit interfaces <i>interface-name</i> unit <i>logical-unit-number</i> family tcc]
Description	For Layer 2.5 VPNs using an Ethernet interface as the TCC router, configure the IP address for which the TCC router is proxying. Ethernet TCC is supported on interfaces that carry IPv4 traffic only. Ethernet TCC encapsulation is supported on one-port Gigabit Ethernet, two-port Gigabit Ethernet, four-port Gigabit Ethernet, and four-port Fast Ethernet PICs only. Ethernet TCC is not supported on the T640 routing node.
Usage Guidelines	See “Configure TCC Encapsulation on Interfaces” on page 19.
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.

remote

Syntax	remote (inet-address mac-address) <i>address</i> ;
Hierarchy Level	[edit interfaces <i>interface-name</i> unit <i>logical-unit-number</i> family tcc]
Description	For Layer 2.5 VPNs employing an Ethernet interface as the TCC router, configure the location of the remote router. Ethernet TCC is supported on interfaces that carry IPv4 traffic only. Ethernet TCC encapsulation is supported on one-port Gigabit Ethernet, two-port Gigabit Ethernet, four-port Gigabit Ethernet, and four-port Fast Ethernet PICs only. Ethernet TCC is not supported on the T640 routing node.
Options	inet-address—Configure the IP address of the remote site. mac-address—Configure the MAC address of the remote site.
Usage Guidelines	See “Configure TCC Encapsulation on Interfaces” on page 19.
Required Privilege Level	interface—To view this statement in the configuration. interface-control—To add this statement to the configuration.

remote-site-id

Syntax	remote-site-id <i>remote-site-ID</i> ;
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i> protocols l2vpn site <i>site-name</i> interface <i>interface-name</i>]
Description	Controls the remote interface to which the interface should connect. The order of the interfaces configured for the site determines the default value if you do not explicitly configure the remote site ID. This statement is optional.
Usage Guidelines	See “Configure the Remote Site ID” on page 24.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

route-distinguisher

Syntax	route-distinguisher (<i>ip-address:number</i> <i>as-number:number</i>);
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i>]
Description	Identifier attached to a route that distinguishes to which VPN it belongs. Each routing instance must have a unique distinguisher associated with it. Each route distinguisher is a 6-byte value.
Options	<p><i>as-number:number</i>—<i>as-number</i> is your assigned autonomous system (AS) number (a 2-byte value) and <i>number</i> is any 4-byte value. The AS number can be in the range of 1 through 65,535.</p> <p><i>ip-address:number</i>—<i>ip-address</i> is an IP address in your assigned prefix range (a 4-byte value) and <i>number</i> is any 2-byte value. The IP address can be any globally unique unicast address.</p>
Usage Guidelines	See “Configure the Route Distinguisher” on page 21.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

route-distinguisher-id

Syntax	route-distinguisher-id <i>ip-address</i> ;
Hierarchy Level	[edit routing-options]
Description	When configured, a route distinguisher is automatically assigned to the routing instance. If you configure the route-distinguisher statement in addition to the route-distinguisher-id statement, the value configured for route-distinguisher supersedes the value generated from route-distinguisher-id.
Usage Guidelines	See “Configure the Route Distinguisher” on page 21.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

site

Syntax site *site-name* {
 site-identifier *identifier*;
 interface *interface-name* {
 remote-site-id *remote-site-ID*;
 }
 }

Hierarchy Level [edit routing-instances *routing-instance-name* protocols l2vpn]

Description Specify the site name, site identifier, and interfaces connecting to the site. Allows you to configure a remote site ID for remote sites.

Options interface *interface-name*—Name of the interface.

site-identifier *identifier*—Numerical identifier for the site used as a default reference for the remote site ID.

remote-site-id *remote-site-ID*—(Optional) Control the remote interface to which the interface should connect. The order of the interfaces configured for the site determines the default value if you do not explicitly configure the remote site ID.

site *site-name*—Name of the site.

Usage Guidelines See “Configure the Site” on page 23.

Required Privilege Level routing—To view this statement in the configuration.
 routing-control—To add this statement to the configuration.

site-identifier

Syntax site-identifier *identifier*;

Hierarchy Level [edit routing-instances *routing-instance-name* protocols l2vpn site *site-name*]

Description The numerical identifier for the site used as a default reference for the remote site ID. It is an unsigned 16-bit number greater than zero.

Usage Guidelines See “Configure the Site” on page 23.

Required Privilege Level routing—To view this statement in the configuration.
 routing-control—To add this statement to the configuration.

traceoptions

Syntax	<pre>traceoptions { file <i>filename</i> <replace> <size <i>size</i>> <files <i>number</i>> <nostamp>; flag <i>flag</i> <<i>flag-modifier</i>> <disable>; }</pre>
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i> protocols l2vpn]
Description	Trace traffic flowing through a Layer 2 VPN.
Options	<p>disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as all.</p> <p>file <i>filename</i>—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks.</p> <p>files <i>number</i>—(Optional) Maximum number of trace files. When a trace file named <i>trace-file</i> reaches its maximum size, it is renamed <i>trace-file.0</i>, then <i>trace-file.1</i>, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.</p> <p>If you specify a maximum number of files, you also must specify a maximum file size with the size option.</p> <p>Range: 2 to 1000 Default: 2 files</p> <p>flag <i>flag</i>—Tracing operation to perform. To specify more than one tracing operation, include multiple flag statements.</p> <ul style="list-style-type: none"> ■ all—All Layer 2 VPN tracing options ■ connections—Layer 2 connections (events and state changes) ■ error—Error conditions ■ nlri—Layer 2 advertisements received or sent by means of the Border Gateway Protocol (BGP) ■ route—Routing information ■ topology—Layer 2 VPN topology changes caused by reconfiguration or advertisements received from other PE routers using BGP <p><i>flag-modifier</i>—(Optional) Modifier for the tracing flag. You can specify the following modifier:</p> <ul style="list-style-type: none"> ■ detail—Provide detailed trace information <p>no stamp—(Optional) Do not place timestamp information at the beginning of each line in the trace file. Default: If you omit this option, timestamp information is placed at the beginning of each line of the tracing output.</p> <p>replace—(Optional) Replace an existing trace file if there is one. Default: If you do not include this option, tracing output is appended to an existing trace file.</p>

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named *trace-file* reaches this size, it is renamed *trace-file.0*. When *trace-file* again reaches its maximum size, *trace-file.0* is renamed *trace-file.1* and *trace-file* is renamed *trace-file.0*. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you also must specify a maximum number of trace files with the *files* option.

Syntax: *xk* to specify KB, *xm* to specify MB, or *xg* to specify GB

Range: 10 KB through the maximum file size supported on your system

Default: 1 MB

Usage Guidelines See “Trace Layer 2 VPN Traffic and Operations” on page 25.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

vrf-export

Syntax vrf-export [*policy-name*];

Hierarchy Level [edit routing-instances *routing-instance-name*]

Description How routes are exported from the local PE router’s VRF table (*routing-instance-name.inet.0*) to the remote PE router. If the instance type is vrf, the vrf-export statement is required.

Options You can configure multiple export policies on the PE router.

Usage Guidelines See “Configure Export Policy for the PE Router’s VRF Table” on page 91.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

vrf-import

Syntax vrf-import [*policy-name*];

Hierarchy Level [edit routing-instances *routing-instance-name*]

Description How routes are imported into the local PE router’s VRF table (*routing-instance-name.inet.0*) from the remote PE router. If the instance type is vrf, the vrf-import statement is required.

Options You can configure multiple import policies on the PE router.

Usage Guidelines See “Configure Import Policy for the PE Router’s VRF Table” on page 90.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Part 3

Layer 3 VPNs

- Layer 3 VPN Overview on page 59
- Layer 3 VPN Configuration Guidelines on page 79
- Layer 3 VPN Configuration Troubleshooting Guidelines on page 113
- Layer 3 VPN Configuration Examples on page 127
- Layer 3 VPN Internet Access Examples on page 215
- Summary of Layer 3 VPN Configuration Statements on page 253

.....

Chapter 6

Layer 3 VPN Overview

The JUNOS software implements Layer 3 Border Gateway Protocol/Multiprotocol Label Switching (BGP/MPLS) virtual private networks (VPNs) as defined in Request for Comments (RFC) 2547 and Internet draft *BGP/MPLS VPNs*, draft-rosen-rfc2547bis (also referred to as RFC 2547bis). This chapter discusses the following topics that provide background information about Layer 3 VPNs:

- Layer 3 VPN Overview on page 59
- Layer 3 VPN Standards on page 60
- Layer 3 VPN Attributes on page 60
- VPN-IPv4 Addresses and Route Distinguishers on page 61
- IPv6 Layer 3 VPNs on page 64
- VPN Routing and Forwarding Tables on page 65
- Route Distribution within a Layer 3 VPN on page 68
- Forwarding across the Provider's Core Network on page 72
- Routing Instances for VPNs on page 73
- Multicast over Layer 3 VPNs on page 73

Layer 3 VPN Overview

In JUNOS, Layer 3 VPNs are based on RFC 2547bis. RFC 2547bis defines a mechanism by which service providers can use their IP backbones to provide VPN services to their customers. A Layer 3 VPN is a set of sites that share common routing information and whose connectivity is controlled by a collection of policies. The sites that make up a Layer 3 VPN are connected over a provider's existing public Internet backbone.

RFC 2547bis VPNs are also known as BGP/MPLS VPNs because BGP is used to distribute VPN routing information across the provider's backbone, and MPLS is used to forward VPN traffic across the backbone to remote VPN sites.

Customer networks, because they are private, can use either public addresses or private addresses, as defined in RFC 1918. When customer networks that use private addresses connect to the public Internet infrastructure, the private addresses might overlap with the same private addresses used by other network users. MPLS/BGP VPNs solve this problem by prefixing a VPN identifier to each address from a particular VPN site, thereby creating an address that is unique both within the VPN and within the public Internet. In addition, each VPN has its own VPN-specific routing table that contains the routing information for that VPN only.

Layer 3 VPN Standards

Layer 3 VPNs are defined in the following documents:

- RFC 2547, *BGP/MPLS VPNs*
- *BGP/MPLS VPNs*, Internet draft draft-rosen-rfc2547bis
- RFC 2283, *Multiprotocol Extensions for BGP4*
- *BGP-MPLS VPN extension for IPv6 VPN over an IPv4 infrastructure*, Internet draft draft-ietf-ppvpn-bgp-ipv6-vpn-02.txt

To access Internet RFCs and drafts, go to the Internet Engineering Task Force (IETF) Web site at <http://www.ietf.org>.

Layer 3 VPN Attributes

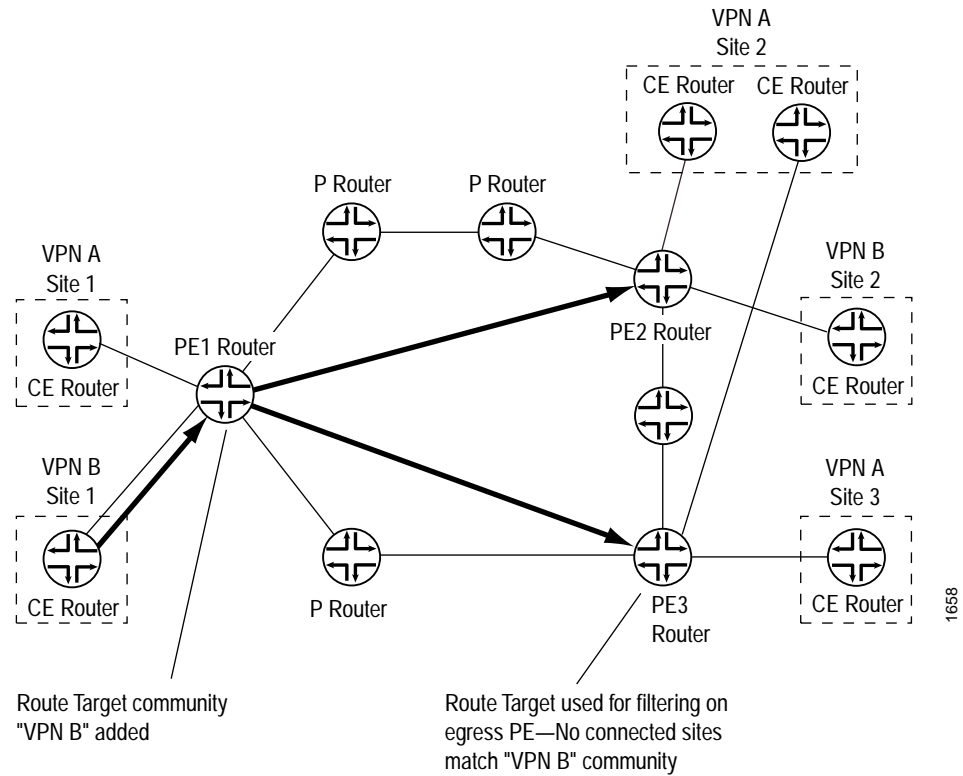
Route distribution within a VPN is controlled using BGP extended community attributes. RFC 2547 defines the following three attributes used by VPNs:

- **Target VPN**—Identifies a set of sites within a VPN to which a provider edge (PE) router distributes routes. This attribute is also called the *route target*. The route target is used by the egress PE router to determine whether a received route is destined for a VPN that the router services.

Figure 4 illustrates the function of the route target. PE Router PE1 adds the route target “VPN B” to routes received from the customer edge (CE) router at Site 1 in VPN B. When it receives the route, the egress router PE2 examines the route target, determines that the route is for a VPN that it services, and accepts the route. When the egress router PE3 receives the same route, it does not accept the route because it does not service any CE routers in VPN B.

- **VPN of origin**—Identifies a set of sites and the corresponding route as having come from one of the sites in that set.
- **Site of origin**—Uniquely identifies the set of routes that a PE router learned from a particular site. This attribute ensures that a route learned from a particular site through a particular PE-CE connection is not distributed back to the site through a different PE-CE connection. It is particularly useful if you are using BGP as the routing protocol between the PE and CE routers and if different sites in the VPN have been assigned the same autonomous system (AS) numbers.

Figure 4: VPN Attributes and Route Distribution

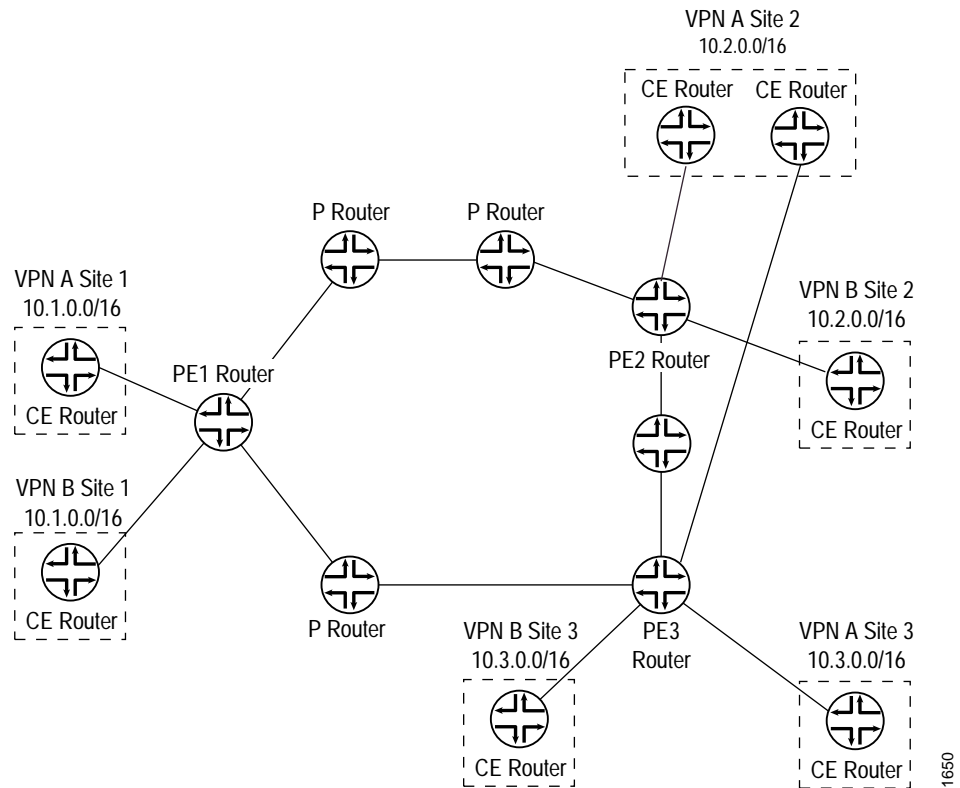


VPN-IPv4 Addresses and Route Distinguishers

Because Layer 3 VPNs connect private networks—which can use either public addresses or private addresses, as defined in RFC 1918—over the public Internet infrastructure, when the private networks use private addresses, the addresses might overlap with the addresses of another private network.

Figure 6 illustrates how private addresses of different private networks can overlap. Here, sites within VPN A and VPN B use the address spaces 10.1.0.0/16, 10.2.0.0/16, and 10.3.0.0/16 for their private networks.

Figure 5: Overlapping Addresses among Different VPNs



To avoid overlapping private addresses, you can configure the network devices to use public addresses instead of private addresses. However, this is a large and complex undertaking. The solution provided in RFC 2547bis uses the existing private network numbers to create a new address that is unambiguous. The new address is part of the VPN-Internet Protocol Version 4 (IPv4) address family, which is a BGP address family added as an extension to the BGP protocol. In VPN-IPv4 addresses, a value that identifies the VPN, called a route distinguisher, is prefixed to the private IPv4 address, providing an address that uniquely identifies a private IPv4 address.

Only the PE routers need to support the VPN-IPv4 address extension to BGP. When an ingress PE router receives an IPv4 route from a device within a VPN, it converts it into a VPN-IPv4 route by prefixing the route distinguisher to the route. The VPN-IPv4 addresses are used only for routes exchanged between PE routers. When an egress PE router receives a VPN-IPv4 route, it converts it back to an IPv4 route by removing the route distinguisher before announcing the route to its connected CE routers.

VPN-IPv4 addresses have the following format:

- Route distinguisher is a 6-byte value that you can specify in one of the following formats:
 - *as-number:number*, where *as-number* is an AS number (a 2-byte value) and *number* is any 4-byte value. The AS number can be in the range 1 through 65,535. We recommend that you use an Internet Assigned Numbers Authority (IANA)-assigned, nonprivate AS number, preferably the Internet service provider's (ISP's) own or the customer's own AS number.
 - *ip-address:number*, where *ip-address* is an IP address (a 4-byte value) and *number* is any 2-byte value. The IP address can be any globally unique unicast address. We recommend that you use the address that you configure in the router-id statement, which is a nonprivate address in your assigned prefix range.
- IPv4 address—4-byte address of a device within the VPN.

Figure 5 illustrates how the AS number can be used in the route distinguisher. Suppose that VPN A is in AS 65535 and that VPN B is in AS 666 (both these AS numbers belong to the ISP), and suppose that the route distinguisher for Site 2 in VPN A is 65535:02 and that the route distinguisher for Site 2 in VPN B is 666:01. When Router PE2 receives a route from the CE router in VPN A, it converts it from its IP address of 10.2.0.0 to a VPN-IPv4 address of 65535:02:10.2.0.0. When the PE router receives a route from VPN B, which uses the same address space as VPN A, it converts it to a VPN-IPv4 address of 666:02:10.2.0.0.

If the IP address is used in the route distinguisher, suppose Router PE2's IP address is 172.168.0.1. When the PE router receives a route from VPN A, it converts it to a VPN-IPv4 address of 172.168.0.1:0:10.2.0.0/16, and it converts a route from VPN B to 172.168.0.0:1:10.2.0.0/16.

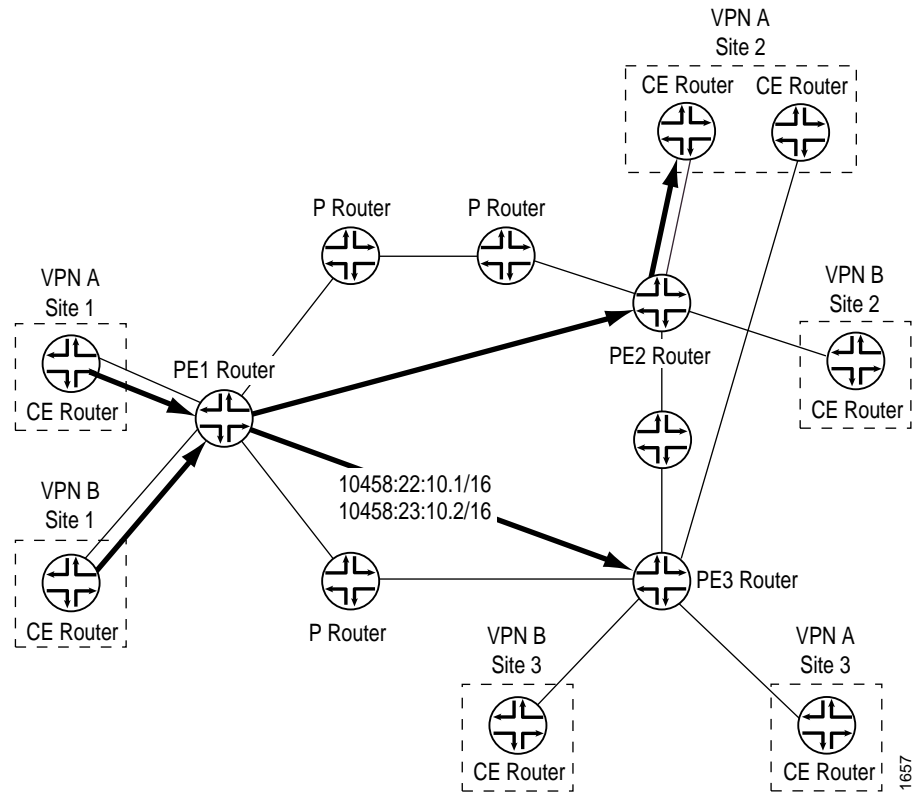
Route distinguishers are used only among PE routers to disambiguate IPv4 addresses from different VPNs. The ingress PE router creates a route distinguisher and converts IPv4 routes received from CE routers into VPN-IPv4 addresses. The egress PE routers convert VPN-IPv4 routes into IPv4 routes before announcing them to the CE router.

Because VPN-IPv4 addresses are a type of BGP address, you must configure internal Border Gateway Protocol (IBGP) sessions between pairs of PE routers so that the PE routers can distribute VPN-IPv4 routes within the provider's core network. (All PE routers are assumed to be within the same AS.)

You define BGP communities to constrain the distribution of routes among the PE routers. Defining BGP communities does not, by itself, disambiguate IPv4 addresses.

Figure 6 illustrates how Router PE1 adds the route distinguisher 10458:22:10.1/16 to routes received from the CE router at Site 1 in VPN A and forwards these routes to the other two PE routers. Similarly, Router PE1 adds the route distinguisher 10458:23:10.2/16 to routes received by the CE router at Site 1 in VPN B and forwards these routes to the other PE routers.

Figure 6: Route Distinguishers



IPv6 Layer 3 VPNs

The interfaces between the PE and CE routers of a Layer 3 VPN can be configured to carry Internet Protocol version 6 (IPv6) traffic. IPv6 is the new version of the Internet Protocol (IP). IP allows numerous nodes on different networks to interoperate seamlessly. Internet Protocol version 4 (IPv4) is currently used in intranets and private networks, as well as the Internet. IPv6 is the successor to IPv4, and is based for the most part on IPv4.

In the Juniper Networks implementation of IPv6, the service provider implements an MPLS-enabled IPv4 backbone to provide VPN service for IPv6 customers. The PE routers have both IPv4 and IPv6 capabilities. They maintain IPv6 VRFs for their IPv6 sites and encapsulate IPv6 traffic in MPLS frames that are then sent into the MPLS core network.

IPv6 for Layer 3 VPNs is supported for BGP and for static routes.

IPv6 over Layer 3 VPNs is described in the Internet draft *BGP-MPLS VPN extension for IPv6 VPN over an IPv4 infrastructure*, draft-ietf-ppvpn-bgp-ipv6-vpn-02.txt.

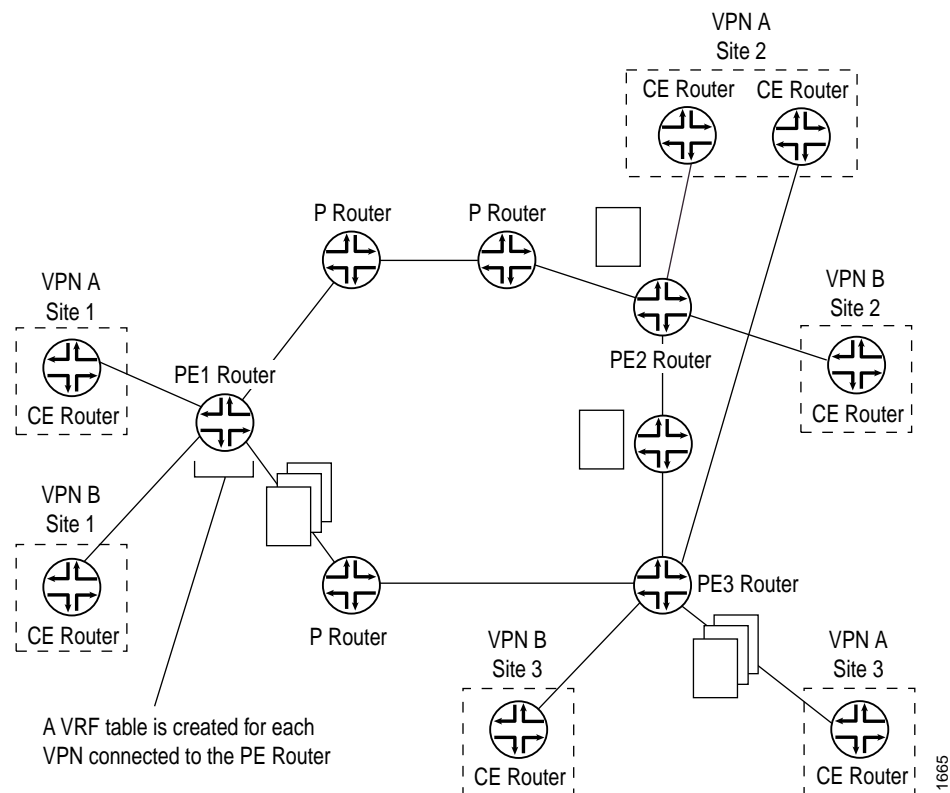
For more information about IPv6, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

VPN Routing and Forwarding Tables

To separate a VPN's routes from routes in the public Internet or those in other VPNs, the PE router creates a separate routing table for each VPN, called a VPN routing and forwarding (VRF) table. The PE router creates one VRF table for each VPN that has a connection to a CE router. Any customer or site that belongs to the VPN can access only the routes in the VRF tables for that VPN.

Figure 7 illustrates the VRF tables that are created on the PE routers. The three PE routers have connections to CE routers that are in two different VPNs, so each PE router creates two VRF tables, one for each VPN.

Figure 7: VRF Tables



Each VRF table is populated from routes received from directly connected CE sites associated with that VRF and from routes received from other PE routers that passed BGP community filtering and are in the same VPN.

Each PE router also maintains one global routing table (inet.0) to reach other routers in and outside the provider's core network.

Each customer connection (that is, each logical interface) is associated with one VRF table. Only the VRF table associated with a customer site is consulted for packets from that site.

You can configure the router so that if a next hop to a destination is not found in the VRF table, the router performs a lookup in the global routing table, which is used for Internet access.

The JUNOS software uses the following routing tables for VPNs:

- **bgp.l3vpn.0**—Stores all VPN-IPv4 unicast routes received from other PE routers. (This table does not store routes received from directly connected CE routers.) This table is present only on PE routers.

When a PE router receives a route from another PE router, it places the route into its **bgp.l3vpn.0** routing table. The route is resolved using the information in the **inet.3** routing table. The resultant route is converted into IPv4 format and redistributed to all *routing-instance-name.inet.0* routing tables on the PE router if it matches the VRF import policy.

The **bgp.l3vpn.0** table is also used to resolve routes over the MPLS tunnels that connect the PE routers. These routes are stored in the **inet.3** routing table. PE-to-PE router connectivity must exist in **inet.3** (not just in **inet.0**) for VPN routes to be resolved properly.

To determine whether to add a route to the **bgp.l3vpn.0** routing table, the JUNOS Internet software checks it against the VRF import policies for all the VPNs configured on the PE router. If the VPN-IPv4 route matches one of the policies, it is added to the **bgp.l3vpn.0** table. To display the routes in the **bgp.l3vpn.0** routing table, use the **show route table bgp.l3vpn.0** command.

- **routing-instance-name.inet.0**—Stores all unicast IPv4 routes received from directly connected CE routers in a routing instance (that is, in a single VPN) and all explicitly configured static routes in the routing instance. This is the VRF table and is present only on PE routers. For example, for a routing instance named VPN-A, the routing table for that instance is named **VPN-A.inet.0**.

When a CE router advertises to a PE router, the PE router places the route into the corresponding *routing-instance-name.inet.0* routing table and advertises the route to other PE routers if it passes a VRF export policy. Among other things, this policy tags the route with the route distinguisher (route target) that corresponds to the VPN site to which the CE belongs. A label is also allocated and distributed with the route. The **bgp.l3vpn.0** routing table is not involved in this process.

The *routing-instance-name.inet.0* table also stores routes announced by a remote PE router that match the VRF import policy for that VPN. The remote PE router redistributed these routes from its **bgp.l3vpn.0** table.

Routes are not redistributed from the *routing-instance-name.inet.0* table to the **bgp.l3vpn.0** table; they are directly advertised to other PE routers.

For each *routing-instance-name.inet.0* routing table, one forwarding table is maintained in the router's Packet Forwarding Engine. This table is maintained in addition to the forwarding tables that correspond to the router's **inet.0** and **mpls.0** routing tables. As with the **inet.0** and **mpls.0** routing tables, the best routes from the *routing-instance-name.inet.0* routing table are placed into the forwarding table.

To display the routes in the *routing-instance-name.inet.0* table, use the **show route table routing-instance-name.inet.0** command.

- **inet.3**—Stores all MPLS routes learned from Label Distribution Protocol (LDP) and Resource Reservation Protocol (RSVP) signaling done for VPN traffic. The routing table stores the MPLS routes only if the `traffic-engineering bgp-igp` option is not enabled.

For VPN routes to be resolved properly, the `inet.3` table must contain routes to all the PE routers in the VPN.

To display the routes in the `inet.3` table, use the `show route inet.3` command.

Note that interior gateway protocol (IGP) shortcuts do not work in VPN environments and should not be configured. IGP shortcuts move routes in `inet.3` to `inet.0`. VPN IBGP (family `inet-vpn`) relies on next-hops that are in the `inet.3` table; thus, IGP shortcuts are incompatible with VPNs.

- **inet.0**—Stores routes learned by the IBGP sessions between the PE routers. To provide Internet access to the VPN sites, configure the `routing-instance-name.inet.0` routing table to contain a default route to the `inet.0` routing table.

To display the routes in the `inet.0` table, use the `show route inet.0` command.

The following routing policies, which are defined in VRF import and export statements, are specific to VRF tables.

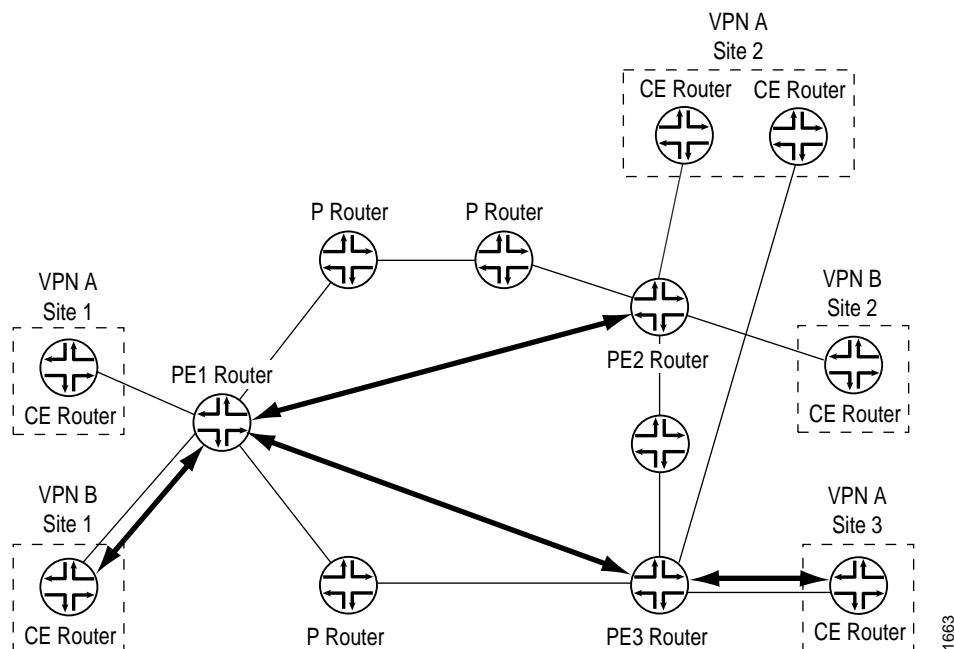
- **Import policy**—Applied to VPN-IPv4 routes learned from another PE router to determine whether the route should be added to the PE router's `bgp.l3vpn.0` routing table. Each routing instance on a PE router has a VRF import policy.
- **Export policy**—Applied to VPN-IPv4 routes that are announced to other PE routers. The VPN-IPv4 routes are IPv4 routes that have been announced by locally connected CE routers.

VPN route processing differs from normal BGP route processing in one way. In BGP, routes are accepted if they are not explicitly rejected by import policy. However, because many more VPN routes are expected, the JUNOS software does not accept (and hence store) VPN routes unless the route matches at least one VRF import policy. If no VRF import explicitly accepts the route, it is discarded and not even stored in the `bgp.l3vpn.0` table. As a result, if a VPN change occurs on a PE router—such as adding a new VRF table or changing a VRF import policy—the PE router sends a BGP route refresh message to the other PE routers (or to the route reflector if this is part of the VPN topology) to retrieve all VPN routes so they can be reevaluated to determine whether they should be kept or discarded.

Route Distribution within a Layer 3 VPN

Within a VPN, the distribution of VPN-IPv4 routes occurs between the PE and CE routers and between the PE routers (see Figure 8).

Figure 8: Route Distribution within a VPN



This section discusses the following:

- Distribution of Routes from CE to PE Routers on page 69
- Distribution of Routes between PE Routers on page 70
- Distribution of Routes from PE to CE Routers on page 71

Distribution of Routes from CE to PE Routers

A CE router announces its routes to the directly connected PE router. The announced routes are in IPv4 format. The PE router places the routes into the VRF table for the VPN. In the JUNOS software, this is the *routing-instance-name.inet.0* routing table, where *routing-instance-name* is the configured name of the VPN.

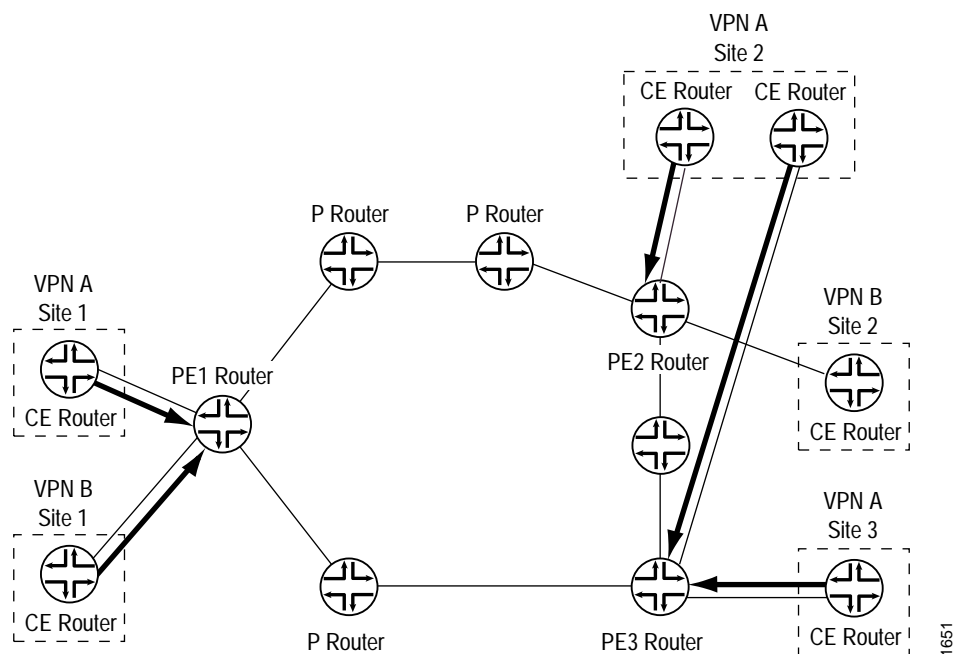
The connection between the CE and PE routers can be a remote connection (a WAN connection) or a direct connection (such as a Frame Relay or Ethernet connection).

CE routers can communicate with PE routers using one of the following:

- Open Shortest Path First (OSPF)
- Routing Information Protocol (RIP)
- BGP
- Static route

Figure 9 illustrates how routes are distributed from CE routers to PE routers. Router PE1 is connected to two CE routers that are in different VPNs. Therefore, it creates two VRF tables, one for each VPN. The CE routers announce IPv4 routes. The PE router installs these routes into two different VRF tables, one for each VPN. Similarly, Router PE2 creates two VRF tables into which routes are installed from the two directly connected CE routers. Router PE3 creates one VRF table because it is directly connected to only one VPN.

Figure 9: Distribution of Routes from CE Routers to PE Routers



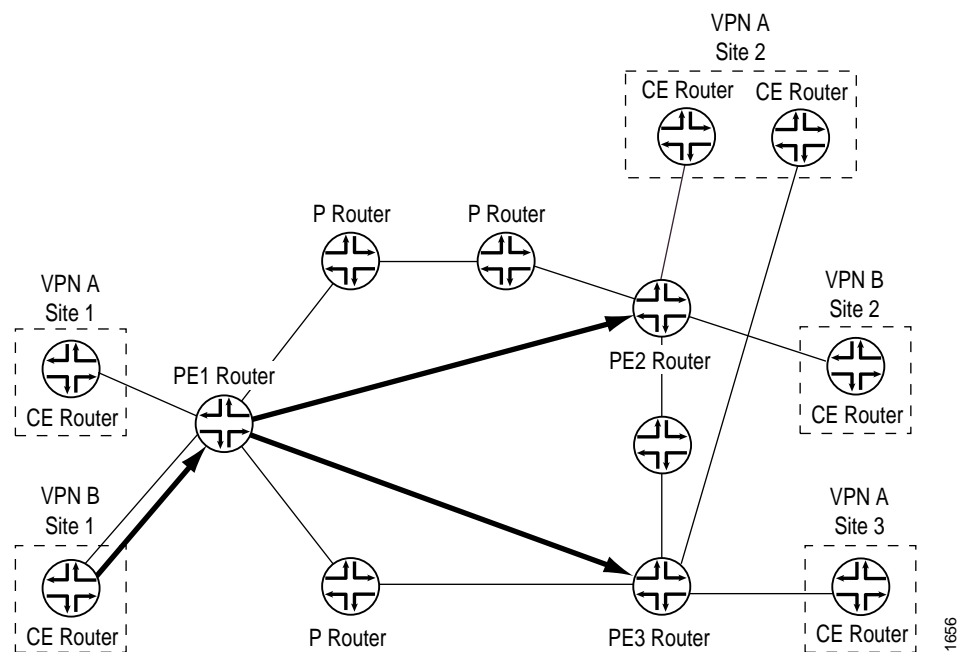
Distribution of Routes between PE Routers

When one PE router receives routes advertised from a directly connected CE router, it checks the received route against the VRF export policy for that VPN. If it matches, the route is converted to VPN-IPv4 format—that is, the route distinguisher (route target) is added to the route. The PE router then announces the route in VPN-IPv4 format to the remote PE routers. The routes are distributed using IBGP sessions, which are configured in the provider's core network. If the route does not match, it is not exported to other PE routers, but can still be used locally for routing, for example, if two CE routers in the same VPN are directly connected to the same PE router.

The remote PE router places the route into its `bgp.l3vpn.0` table if the route passes the import policy on the IBGP session between the PE routers. At the same time, it checks the route against the VRF import policy for the VPN. If it matches, the route distinguisher is removed from the route and it is placed into the VRF table (the `routing-instance-name.inet.0` table) in IPv4 format.

Figure 10 illustrates how Router PE1 distributes routes to the other PE routers in the provider's core network. Router PE2 and Router PE3 each have VRF import policies that they use to determine whether to accept routes received over the IBGP sessions and install them in their VRF tables.

Figure 10: Distribution of Routes between PE Routers



1656

Distribution of Routes from PE to CE Routers

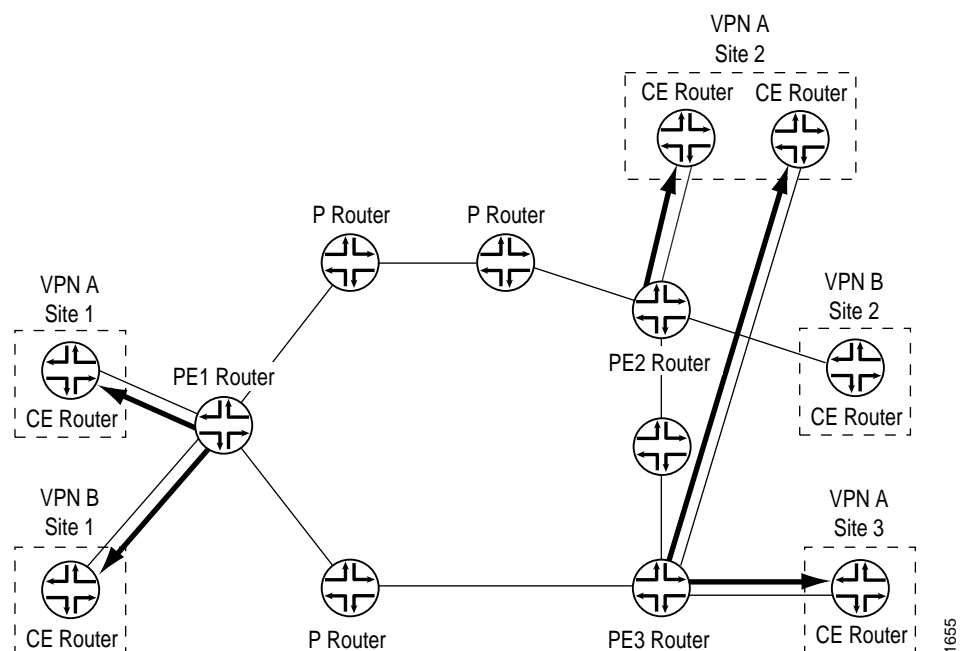
The remote PE router announces the routes in its VRF tables, which are in IPv4 format, to its directly connected CE routers.

PE routers can communicate with CE routers using one of the following routing protocols:

- OSPF
- RIP
- BGP
- Static route

Figure 11 illustrates how the three PE routers announce their routes to their connected CE routers.

Figure 11: Distribution of Routes from PE Routers to CE Routers



Forwarding across the Provider's Core Network

The PE routers in the provider's core network are the only routers that are configured to support VPNs and hence are the only routers that know about the existence of the VPNs. From the point of view of VPN functionality, the provider routers in the core—those provider routers that are not directly connected to CE routers—are merely routers along the tunnel between the ingress and egress PE routers.

The tunnels can be either LDP or MPLS. Any provider routers along the tunnel must support the protocol used for the tunnel, either LDP or MPLS.

When PE-router-to-PE router forwarding is tunneled over MPLS label-switched paths (LSPs), the MPLS packets have a two-level label stack (see Figure 12):

- Outer label—Label assigned to the address of the BGP next hop by the IGP next hop
- Inner label—Label that the BGP next hop assigned for the packet's destination address

Figure 12: Using MPLS LSPs to Tunnel between PE Routers

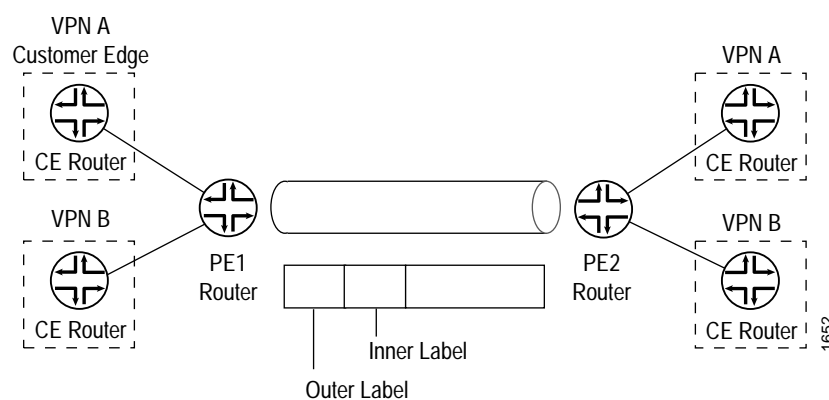
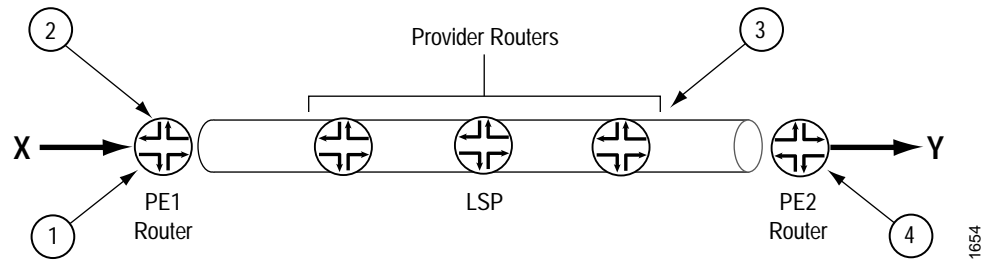


Figure 13 illustrates how the labels are assigned and removed:

1. When CE Router X forwards a packet to Router PE1 with a destination of CE Router Y, the PE route identifies the BGP next hop to Router Y and assigns a label that corresponds to the BGP next hop and identifies the destination CE router. This label is the inner label.
2. Router PE1 then identifies the IGP route to the BGP next hop and assigns a second label that corresponds to the LSP of the BGP next hop. This label is the outer label.
3. The inner label remains the same as the packet traverses the LSP tunnel. The outer label is swapped at each hop along the LSP and is then popped by the penultimate hop router (the third provider router).
4. Router PE2 pops the inner label from the route and forwards the packet to Router Y.

Figure 13: Label Stack



Routing Instances for VPNs

To implement Layer 3 VPNs in the JUNOS software, you configure one routing instance for each VPN. You configure the routing instances on PE routers only. Each VPN routing instance consists of the following components:

- VRF table—On each PE router, you configure one VRF table for each VPN.
- Set of interfaces that use the VRF table—The logical interface to each directly connected CE router must be associated with a VRF table. You can associate more than one interface with the same VRF table if more than one CE router in a VPN is directly connected to the PE router.
- Policy rules—These control the import of routes into and the export of routes from the VRF table.
- One or more routing protocols that install routes from CE routers into the VRF table—You can use the BGP, OSPF, and RIP routing protocols, and you can use static routes.

Multicast over Layer 3 VPNs

You can configure multicast routing over a network running a Layer 3 VPN that complies with RFC 2547. This section describes this type of network application and includes these topics:

- Multicast over Layer 3 VPNs Overview on page 74
- Sending PIM Hello Messages to the PE Routers on page 75
- Sending PIM Join Messages to the PE Routers on page 76
- Receiving the Multicast Transmission on page 77

Multicast over Layer 3 VPNs Overview

In the unicast environment of a Layer 3 VPN, all VPN state information is contained within the PE routers. In a multicast Layer 3 VPN environment, Protocol Independent Multicast (PIM) adjacencies are established between the CE router and the PE router and between the master PIM instance. They are configured at the [protocols pim] hierarchy level on the IGP neighbors of the PE router. The set of master PIM adjacencies on the service provider's network make up the forwarding path, which consists of a rendezvous point (RP) tree rooted at the RP within the service provider's network.

Therefore, provider (P) routers within the provider network must maintain multicast state information for the Layer 3 VPNs. For this to function, there must be two types of rendezvous points for each VPN:

- The VPN-RP, an RP that resides within the VPN
- The service provider RP (SP-RP), which resides within the service provider network

A PE router can act as an SP-RP, but cannot be the VPN-RP of a Layer 3 VPN. The VPN-RP must be located on a CE router or some other customer router within the VPN.

To configure multicast over a Layer 3 VPN, you must install a Tunnel Services Physical Interface Card (PIC) on the following devices:

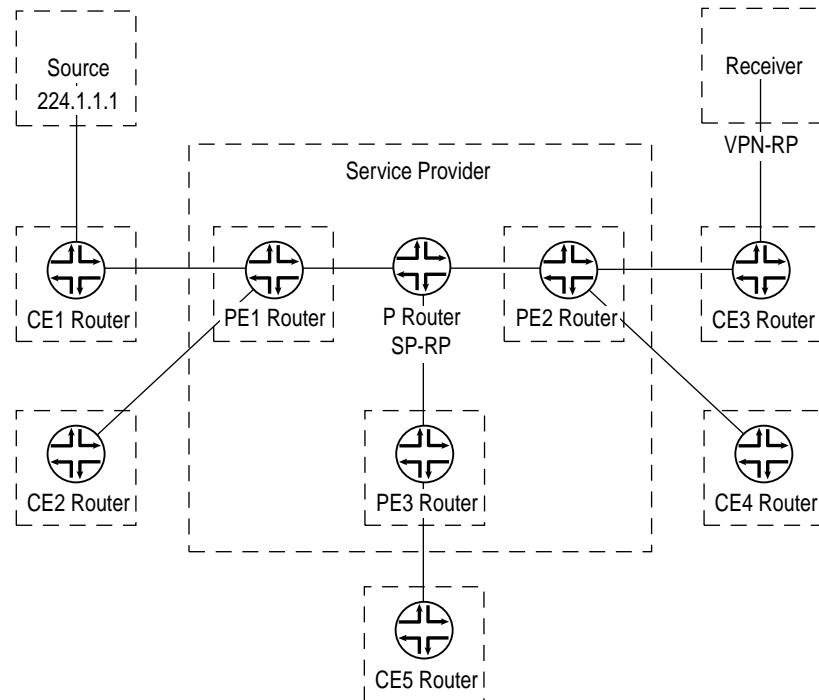
- Provider routers acting as rendezvous points
- PE routers configured to run multicast routing
- CE routers acting as destination routers or as VPN-RPs

For more information about running multicast over Layer 3 VPNs, see the following documents:

- *Multicast in MPLS/BGP VPNs*, Internet draft draft-rosen-vpn-mcast-02.txt
- *JUNOS Internet Software Configuration Guide: Multicast*

The sections that follow describe the operation of a multicast VPN. Figure 14 illustrates the network topology used.

Figure 14: Multicast Topology Overview



1743

Sending PIM Hello Messages to the PE Routers

The first step in initializing multicast over a Layer 3 VPN is the distribution of a PIM Hello message from a PE router (called PE3 in this section) to all the other PE routers on which PIM is configured.

You configure PIM on the Layer 3 VPN routing instance on the PE3 router. If a Tunnel PIC exists on the router, a multicast interface is created. This interface is used to communicate between the PIM instance within the VRF and the master PIM instance.

The following occurs when a PIM Hello message is sent to the PE routers:

1. A PIM Hello message is sent from the VRF over the multicast interface. A generic routing encapsulation (GRE) header is prepended to the PIM Hello message. The header message includes the VPN group address and the loopback address of the PE3 router.
2. A PIM register header is prepended to the Hello message as the packet is looped through the PIM encapsulation interface. This header contains the destination address of the SP-RP and the loopback address of the PE3 router.
3. The packet is sent to the SP-RP.

4. The SP-RP removes the top header from the packet and sends the remaining GRE-encapsulated Hello message to all the PE routers.
5. The master PIM instance on each PE router handles the GRE encapsulated packet. Because the VPN group address is contained in the packet, the master instance removes the GRE header from the packet and sends the Hello message, which contains the proper VPN group address within the VRF, over the multicast interface.

Sending PIM Join Messages to the PE Routers

To receive a multicast broadcast from a multicast network, a CE router must send a PIM Join message to the VPN-RP. The process described in this section refers to Figure 14.

The CE5 router needs to receive a multicast broadcast from multicast source 224.1.1.1. To receive the broadcast, it sends a PIM Join message to the VPN-RP (the PE3 router):

1. The PIM Join message is sent through the multicast interface, and a GRE header is prepended to the message. The GRE header contains the VPN group ID and the loopback address of the PE3 router.
2. The PIM Join message is then sent through the PIM encapsulation interface and a register header is prepended to the packet. The register header contains the IP address of the SP-RP and the loopback address of the PE3 router.
3. The PIM Join message is sent to the SP-RP by means of unicast routing.
4. On the SP-RP, the register header is stripped off (the GRE header remains) and the packet is sent to all the PE routers.
5. The PE2 router receives the packet, and because the link to the VPN-RP is through the PE2 router, it sends the packet through the multicast interface to remove the GRE header.
6. Finally, the PIM Join message is sent to the VPN-RP.

Receiving the Multicast Transmission

The steps that follow outline how a multicast transmission is propagated across the network:

1. The multicast source connected to the CE1 router sends the packet to group 224.1.1.1 (the VPN group address). The packet is encapsulated into a PIM register.
2. Because this packet already includes the PIM header, it is forwarded by means of unicast routing to the VPN-RP over the Layer 3 VPN.
3. The VPN-RP removes the packet and sends it out the downstream interfaces (which include the interface back to the CE3 router). The CE3 router also forwards this to the PE3 router.
4. The packet is sent through the multicast interface on the PE2 router; in the process, the GRE header is prepended to the packet.
5. Next, the packet is sent through the PIM encapsulation interface, where the register header is prepended to the data packet.
6. The packet is then forwarded to the SP-RP, which removes the register header, leaves the GRE header intact, and sends the packet to the PE routers.
7. PE routers remove the GRE header and forward the packet to the CE routers that requested the multicast broadcast by sending the PIM Join message.



PE routers that have not received requests for multicast broadcasts from their connected CE routers still receive packets for the broadcast. These PE routers drop the packets as they are received.

.....

Chapter 7

Layer 3 VPN Configuration Guidelines

To configure Layer 3 virtual private network (VPN) functionality, you must enable VPN support on the provider edge (PE) router. You must also configure any provider (P) routers that service the VPN, and you must configure the customer edge (CE) routers so that their routes are distributed into the VPN.

To configure Layer 3 VPNs, you include statements at the [edit routing-instances] hierarchy level:

```
[edit]
routing-instances {
  routing-instance-name {
    description text;
    interface interface-name;
    instance-type vrf;
    route-distinguisher ( as-number:number | ip-address:number );
    vrf-import [ policy-names ];
    vrf-export [ policy-names ];
    vrf-table-label;
    protocols {
      bgp {
        bgp-configuration;
      }
      ospf {
        ospf-configuration;
      }
      pim {
        pim-configuration;
        vpn-group-address address;
      }
      rip {
        rip-configuration;
      }
    }
  }
  routing-options {
    autonomous-system autonomous-system <loops number>;
    forwarding-table {
      export [ policy-names ];
    }
    interface-routes {
      rib-group group-name;
    }
    martians {
      destination-prefix match-type <allow>;
    }
    maximum-routes route-limit <log-only | threshold value>;
```

```

options {
    syslog (level level | upto level);
}
rib routing-table {
    static {
        defaults {
            static-options;
        }
        route destination-prefix {
            next-hop;
            static-options;
        }
    }
}
martians {
    destination-prefix match-type <allow>;
}
static {
    defaults {
        static-options;
    }
    route destination-prefix {
        policy [ policy-names ];
        static-options;
    }
}
}
router-id address;
static {
    defaults {
        static-options;
    }
    route destination-prefix {
        policy [ policy-names ];
        static-options;
    }
}
}
}
}

```

For Layer 3 VPNs, only some of the statements in the [edit routing-instances] hierarchy are valid. For the full hierarchy, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

In addition to these statements, you must enable a signaling protocol, internal Border Gateway Protocol (IBGP) sessions between the PE routers, and an interior gateway protocol (IGP) on the PE and provider routers.

By default, Layer 3 VPNs are disabled.

This chapter describes the following tasks for configuring VPNs:

- Enable a Signaling Protocol on page 81
- Configure an IGP on PE and Provider Routers on page 84
- Configure an IBGP Session between PE Routers on page 85
- Configure Routing Instances for Layer 3 VPNs on PE Routers on page 85
- Configure VPN Routing between the PE and CE Routers on page 96
- Configure VPN Graceful Restart on page 105
- Configure Multicast over Layer 3 VPNs on page 105
- Configure a GRE Tunnel Interface for Layer 3 VPNs on page 106
- Configure an ES Tunnel Interface for Layer 3 VPNs on page 108
- Configure IPsec between PE Routers Instead of MPLS on page 110
- Configure Packet Forwarding for VPNs on page 112

For configuration examples, see “Layer 3 VPN Configuration Examples” on page 127.

Enable a Signaling Protocol

For Layer 3 VPNs to function, you must enable a signaling protocol on the PE routers. You can do one of the following:

- Use LDP for VPN Signaling on page 81
- Use RSVP for VPN Signaling on page 83

Use LDP for VPN Signaling

To use Label Distribution Protocol (LDP) for VPN signaling, perform the following steps on the PE and provider routers:

1. Configure LDP on the interfaces in the core of the service provider's network by including the `ldp` statement at the `[edit protocols]` hierarchy level. You need to configure LDP only on the interfaces between PE routers or between PE and provider routers. You can think of these as the “core-facing” interfaces. You do not need to configure LDP on the interface between the PE and CE routers.

```
[edit]
protocols {
  ldp {
    interface interface-name;
  }
}
```

2. Configure the Multiprotocol Label Switching (MPLS) address family on the interfaces on which you enabled LDP (the interfaces you configured in Step 1):

```
[edit]
interfaces {
  interface-name {
    unit logical-unit-number {
      family mpls;
    }
  }
}
```

Specify the interface name in the format *type-fpc/pic/port*.

3. Configure Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS) on each PE and provider router. You configure these protocols at the master instance of the routing protocol, not within the routing instance used for the VPN.

To configure OSPF, include the `ospf` statement at the `[edit protocols]` hierarchy level. At a minimum, you must configure a backbone area on at least one of the router's interfaces.

```
[edit]
protocols {
  ospf {
    area 0.0.0.0 {
      interface interface-name;
    }
  }
}
```

To configure IS-IS, include the `isis` statement at the `[edit protocols]` hierarchy level and configure the loopback interface and International Organization for Standardization (ISO) family at the `[edit interfaces]` hierarchy level. At a minimum, you must enable IS-IS on the router, configure a network entity title (NET) on one of the router's interfaces (preferably the loopback interface, `lo0`), and configure the ISO family on all interfaces on which you want IS-IS to run. When you enable IS-IS, Level 1 and Level 2 are enabled by default. The following is the minimum IS-IS configuration. In the address statement, *address* is the NET.

```
[edit]
interfaces {
  lo0 {
    unit logical-unit-number {
      family iso {
        address address;
      }
    }
  }
  type-fpc/pic/port {
    unit logical-unit-number {
      family iso;
    }
  }
}
protocols {
  isis {
    interface all;
  }
}
```

For more information about configuring OSPF and IS-IS, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

Use RSVP for VPN Signaling

To use the Resource Reservation Protocol (RSVP) for VPN signaling, perform the following steps:

1. On each PE router, configure traffic engineering. To do this, you must configure an IGP that supports traffic engineering (either IS-IS or OSPF) and enable traffic engineering support for that protocol.

To enable OSPF traffic engineering support, include the `traffic-engineering` statement at the `[edit protocols ospf]` hierarchy level:

```
[edit protocols ospf]
traffic-engineering {
  no-topology;
  shortcuts;
}
```

For IS-IS, traffic engineering support is enabled by default.

2. On each PE and provider router, enable RSVP on the router interfaces that participate in the label-switched path (LSP). On the PE router, these are the interfaces that are the ingress and egress points to the LSP. On the provider router, these are the interfaces that connect the LSP between the PE routers. Do not enable RSVP on the interface between the PE and the CE routers, because this interface is not part of the LSP.

To configure RSVP on the PE and provider routers, include the `interface` statement at the `[edit protocols rsvp]` hierarchy level. Include one `interface` statement for each interface on which you are enabling RSVP.

```
[edit protocols]
rsvp {
  interface interface-name;
  interface interface-name;
}
```

3. On each PE router, configure an MPLS LSP to the PE router that is the LSP's egress point. To do this, include the `label-switched-path` and `interface` statements at the `[edit protocols mpls]` hierarchy level.

```
[edit protocols]
mpls {
  label-switched-path path-name {
    to ip-address;
  }
  interface interface-name;
}
```

In the `to` statement, specify the address of the LSP's egress point, which is an address on the remote PE router.

In the `interface` statement, specify the name of the interface (both the physical and logical portions). Include one `interface` statement for the interface associated with the LSP.

When you configure the same interface at the [edit interfaces] hierarchy level, you must also configure family mpls and family inet when configuring the logical interface:

```
[edit interfaces]
interface-name {
  unit logical-unit-number {
    family inet;
    family mpls;
  }
}
```

4. On all provider routers that participate in the LSP, enable MPLS by including the interface statement at the [edit mpls] hierarchy level. Include one interface statement for each connection to the LSP.

```
[edit]
mpls {
  interface interface-name;
  interface interface-name;
}
```

5. Enable MPLS on the interface between the PE and CE routers by including the interface statement at the [edit mpls] hierarchy level. Doing this allows the PE router to assign an MPLS label to traffic entering the LSP or to remove the label from traffic exiting the LSP.

```
[edit]
mpls {
  interface interface-name;
}
```

For information about configuring MPLS, see the *JUNOS Internet Software Configuration Guide: MPLS Applications*.

Configure an IGP on PE and Provider Routers

To allow the PE and provider routers to exchange routing information, you must either configure an IGP on all the routers or you must configure static routes. You configure the IGP on the master instance of the routing protocol process (rpd) (that is, at the [edit protocols] hierarchy level), not within the routing instance used for the VPN (that is, not at the [edit routing-instances] hierarchy level).

When you configure the PE router, do not configure any summarization of the PE router's loopback addresses at the area boundary. Each PE router's loopback address should appear as a separate route.

For information about configuring routing protocols and static routes, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

Configure an IBGP Session between PE Routers

You must configure an IBGP session between PE routers to allow the PE routers to exchange information about routes originating and terminating in the VPN. To do this, include the family inet-vpn statement when configuring IBGP:

```
[edit protocols]
bgp {
  group group-name {
    type internal;
    local-address ip-address;
    family inet-vpn {
      unicast;
    }
    neighbor ip-address;
  }
}
```

The family inet-vpn statement indicates that the IBGP session is for the VPN.

The IP address in the local-address statement is the address of the loopback interface (lo0) on the local PE router. The IBGP session for VPNs runs through the loopback address. (You must also configure the lo0 interface at the [edit interfaces] hierarchy level.)

The IP address in the neighbor statement is the loopback address of the neighboring PE router. If you are using RSVP signaling, this IP address is the same address you specify in the to statement at the [edit mpls label-switched-path *lsp-path-name*] hierarchy level when you configure the MPLS LSP.

Configure Routing Instances for Layer 3 VPNs on PE Routers

To configure routing instances for Layer 3 VPNs, include the routing-instances statement at the [edit] hierarchy level. You configure VPN routing instances only on PE routers.

```
[edit]
routing-instances {
  routing-instance-name {
    description text;
    interface interface-name;
    instance-type vrf;
    route-distinguisher ( as-number:number | ip-address:number );
    vrf-import [ policy-name ];
    vrf-export [ policy-name ];
    vrf-target {
      export community-name;
      import community-name;
    }
  }
}
```



Note

For the VPN to function, you must include the instance-type, interface, route-distinguisher, vrf-import, and vrf-export statements in the routing instance configuration on the PE router. The vrf-table-label statement is optional.

The following sections describe how to configure VPN routing instances:

- Configure the Description on page 86
- Configure the Instance Type on page 86
- Configure Interfaces for VPN Routing on page 86
- Configure a Logical Unit on the Loopback Interface on page 88
- Configure the Route Distinguisher on page 89
- Configure Policy for the PE Router's VRF Table on page 89

Configure the Description

To provide a textual description for the routing instance, include the description statement at the [edit routing-instances *routing-instance-name*] hierarchy level. Enclose any descriptive text that includes spaces in quotation marks (" "). Any descriptive text you include is displayed in the output of the show route instance detail command and has no effect on the operation of the routing instance.

```
[edit routing-instances routing-instance-name]
description text;
```

Configure the Instance Type

Each PE router uses a VPN routing and forwarding (VRF) table for distributing routes within the VPN. To create the VRF table on the PE router, include the instance-type statement at the [edit routing-instances *routing-instance-name*] hierarchy level, specifying the instance type as vrf:

```
[edit routing-instances routing-instance-name]
instance-type vrf;
```

Configure Interfaces for VPN Routing

On each PE router, you must configure an interface over which the VPN traffic travels between the PE and CE routers. To do this, include the interface statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit routing-instances routing-instance-name]
interface interface-name;
```

Specify both the physical and logical portions of the interface name, in the following format:

```
physical.logical
```

For example, in so-1/2/1.0, so-1/2/1 is the physical portion of the interface name and 0 is the logical portion. If you do not specify the logical portion of the interface name, 0 is used.

A logical interface can be associated with only one routing instance.

When you configure the same interface at the [edit interfaces] hierarchy level, you must also configure family inet when configuring the logical interface:

```
[edit interfaces]
interface-name {
  unit logical-unit-number {
    family inet;
  }
}
```



Note

If you downgrade JUNOS to version 5.5 or earlier, the VPN configuration might become invalid. VPN interfaces between PE and CE routers formerly required the family mpls statement to be configured.

When you configure carrier-of-carriers VPNs, you need to configure the family mpls statement in addition to the family inet statement for the interfaces between the PE and CE routers. For carrier-of-carriers VPNs, configure the logical interface as follows:

```
[edit interfaces]
interface-name {
  unit logical-unit-number {
    family inet;
    family mpls;
  }
}
```

If you configure family mpls on the logical interface and then configure this interface for a non-carrier-of-carriers vrf routing instance, the family mpls statement is automatically removed from the configuration for the logical interface, since it is not needed.



Note

If you enable a routing protocol on all instances by specifying interfaces all when configuring the master instance of the protocol at the [edit protocols] hierarchy level, and if you configure a specific interface for VPN routing at the [edit routing-instances *routing-instance-name*] hierarchy level, the latter interface statement takes precedence and the interface is used exclusively for the VPN.

If you explicitly configure the same interface name at both the [edit protocols] and [edit routing-instances *routing-instance-name*] hierarchy levels, when you try to commit the configuration, it will fail.

Configure a Logical Unit on the Loopback Interface

You can configure a logical unit on the loopback interface into each VRF routing instance you have configured on the router. This is only possible on Layer 3 VPNs (VRF routing instances). Associating a VRF routing instance with a logical unit on the loopback interface allows you to easily identify the VRF. This is useful for troubleshooting, allowing you to ping a remote CE router from a local PE router in a Layer 3 VPN. See “Ping a Remote CE Router from a PE Router” on page 123 for more information.

You can also configure a firewall filter for the logical unit on the loopback interface, allowing you to filter traffic for the VRF routing instance associated with it.

The following describes how firewall filters affect the VRF routing instance depending on whether they are configured on the default loopback interface, the VRF routing instance, or some combination of the two. Note that “default loopback interface” refers to lo0.0 (associated with the default routing table) and “VRF loopback interface” refers to lo0.*n* which is configured in the VRF routing instance.

- If you configure Filter A on the default loopback interface and Filter B on the VRF loopback interface, the VRF routing instance uses Filter B.
- If you configure Filter A on the default loopback interface, but do not configure a filter on the VRF loopback interface, the VRF routing instance does not use a filter.
- If you configure Filter A on the default loopback interface but do not even configure a VRF loopback interface, the VRF routing instance uses Filter A.

To configure a logical unit on the loopback interface, configure the unit statement at the [edit interfaces lo0] hierarchy level:

```
[edit interfaces]
lo0 {
  unit number {
    family inet {
      address address;
    }
  }
}
```

To associate a firewall filter with the logical unit on the loopback interface, include the following statements at the [edit interfaces lo0 unit *unit-number* family inet] hierarchy level:

```
[edit interfaces lo0 unit unit-number family inet]
filter {
  input filter-name;
}
```

You also need to include the lo0.*n* interface in the configuration for the VRF routing instance at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit]
routing-instances {
  routing-instance-name {
    interface lo0.n;
  }
}
```

For more information on how to configure firewall filters, see the *JUNOS Internet Software Configuration Guide: Policy Framework*.

Configure the Route Distinguisher

The configuration for a route distinguisher in a Layer 3 VPN is identical to the configuration in a Layer 2 VPN. See “Configure the Route Distinguisher” on page 21.

Configure Policy for the PE Router’s VRF Table

On each PE router, you must define policies that define how routes are imported into and exported from the router’s VRF table. In these policies, you must define the route target and you can optionally define the route origin.

The following sections describe how to configure policy for the VRF tables:

- Configure the Route Target on page 89
- Configure the Route Origin on page 90
- Configure Import Policy for the PE Router’s VRF Table on page 90
- Configure Export Policy for the PE Router’s VRF Table on page 91
- Apply Both the VRF Export and the BGP Export Policies on page 92
- Configure a VRF Target on page 93
- Filter Traffic Based on the IP Header on page 94
- Configure a VPN Tunnel for VRF Table Lookup on page 95

Configure the Route Target

In the import and export policies for the PE router’s VRF table, you must define the route target, which defines which VPN the route is part of. To do this, include the target option in the community statement at the [edit policy-options] hierarchy level:

```
[edit policy-options]
community name members target: community-id;
```

name is the name of the community.

community-id is the identifier of the community. You specify it in one of the following formats:

- *as-number:number*, where *as-number* is an autonomous system (AS) number (a 2-byte value) and *number* is a 4-byte community identifier. The AS number can be in the range 1 through 65,535. We recommend that you use an Internet Assigned Numbers Authority (IANA)-assigned, nonprivate AS number, preferably the Internet service provider’s (ISP’s) own or the customer’s own AS number. The community identifier can be a number in the range 0 through $2^{32} - 1$.
- *ip-address:number*, where *ip-address* is an Internet Protocol Version 4 (IPv4) address (a 4-byte value) and *number* is a 2-byte community identifier. The IP address can be any globally unique unicast address. We recommend that you use the address that you configure in the router-id statement, which is a nonprivate address in your assigned prefix range. The community identifier can be a number in the range 1 through 65,535.

Configure the Route Origin

In the import and export policies for the PE router's VRF table, you can optionally define the route origin (also known as the site of origin), which identifies the set of routes learned from a particular CE site. This attribute ensures that a route learned from a particular site through a particular PE-CE connection is not distributed back to the site through a different PE-CE connection. It is particularly useful if you are using the Border Gateway Protocol (BGP) as the routing protocol between the PE and CE routers and if different sites in the VPN have been assigned the same AS numbers.

To configure a route origin, complete the following steps:

1. Include the origin option in the community statement at the [edit policy-options] hierarchy level:

```
[edit policy-options]
community name members origin: community-id;
```

name is the name of the community.

community-id is the identifier of the community. You specify it in one of the following format:

- *as-number:number*, where *as-number* is an AS number (a 2-byte value) and *number* is a 4-byte community identifier. The AS number can be in the range 1 through 65,535. We recommend that you use an IANA-assigned, nonprivate AS number, preferably the ISP's own or the customer's own AS number. The community identifier can be a number in the range 0 through $2^{32} - 1$.
- *ip-address:number*, where *ip-address* is an IPv4 address (a 4-byte value) and *number* is a 2-byte community identifier. The IP address can be any globally unique unicast address. We recommend that you use the address that you configure in the router-id statement, which is a nonprivate address in your assigned prefix range. The community identifier can be a number in the range 1 through 65,535.

2. Include the community in the import policy for the PE router's VRF table by configuring the community statement with the *community-id* defined in Step 1 at the [edit policy-options policy-statement *import-policy-name* term *import-term-name* from] hierarchy level. See "Configure Import Policy for the PE Router's VRF Table" on page 90.
3. Include the community in the export policy for the PE router's VRF table by configuring the community statement with the *community-id* defined in Step 1 at the [edit policy-options policy-statement *export-policy-name* term *export-term-name* then] hierarchy level. See "Configure Export Policy for the PE Router's VRF Table" on page 91.

Configure Import Policy for the PE Router's VRF Table

Each VPN must have a policy that defines how routes are imported into the PE router's VRF table. An import policy is applied to routes received from other PE routers in the VPN. A policy must evaluate all routes received over the IBGP session with the peer PE router. If the routes match the conditions, the route is installed in the PE router's *routing-instance-name.inet.0* VRF table. An import policy must contain a second term that rejects all other routes.

Unless an import policy contains only a then reject statement, it must include a reference to a community. Otherwise, when you try to commit the configuration, the commit fails. Note that you can configure multiple import policies.

An import policy determines what to import to a specified VRF table based on the VPN routes learned from the remote PE routers through IBGP. The IBGP session is configured at the [edit protocols bgp] hierarchy level. If you also configure an import policy at the [edit protocols bgp] hierarchy level, the import policies at the [edit policy-options] hierarchy level and the [edit protocols bgp] hierarchy level are combined through a logical AND operation. This allows you to filter traffic as a group.

To configure an import policy for the PE router's VRF table, follow these steps:

1. To define an import policy, include the policy-statement statement at the [edit policy-options] hierarchy level. For all PE routers, an import policy must always include the following, at a minimum:

```
[edit]
policy-options {
  policy-statement import-policy-name {
    term import-term-name {
      from {
        protocol bgp;
        community community-id;
      }
      then accept;
    }
    term term-name {
      then reject;
    }
  }
}
```

The *import-policy-name* policy evaluates all routes received over the IBGP session with the other PE router. If the routes match the conditions in the from statement, the route is installed in the PE router's *routing-instance-name*.inet.0 VRF table. The second term in the policy rejects all other routes.

For more information about creating policies, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

2. To configure an import policy, include the vrf-import statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit routing-instances routing-instance-name]
vrf-import [ import-policy-name ];
```

Configure Export Policy for the PE Router's VRF Table

Each VPN must have a policy that defines how routes are exported from the PE router's VRF table. An export policy is applied to routes sent to other PE routers in the VPN. An export policy must evaluate all routes received over the routing protocol session with the CE router. (This session can use the BGP, OSPF, or Routing Information Protocol (RIP) routing protocols, or static routes.) If the routes match the conditions, the specified community target (which is the route target) is added to them and they are exported to the remote PE routers. An export policy must contain a second term that rejects all other routes.

Export policies defined within the VPN routing instance are the only export policies that apply to the VRF table. Any export policy that you define on the IBGP session between the PE routers has no effect on the VRF table. Note that you can configure multiple export policies.

To configure an export policy for the PE router's VRF table, follow these steps:

1. To define an export policy, include the `policy-statement` statement at the `[edit policy-options]` hierarchy level. For all PE routers, an export policy must distribute VPN routes to and from the connected CE routers in accordance with the type of routing protocol that you configure between the CE and PE routers within the routing instance. An export policy must always include the following, at a minimum:

```
[edit]
policy-options {
  policy-statement export-policy-name {
    term export-term-name {
      from protocol (bgp | ospf | rip | static);
      then {
        community add community-id;
        accept;
      }
    }
    term term-name {
      then reject;
    }
  }
}
```

The *export-policy-name* policy evaluates all routes received over the routing protocol session with the CE router. (This session can use the BGP, OSPF, or RIP routing protocol, or static routes.) If the routes match the conditions in the `from` statement, the community target specified in the `then community add` statement is added to them and they are exported to the remote PE routers. The second term in the policy rejects all other routes.

For more information about configuring routing within the routing instance, see “Configure VPN Routing between the PE and CE Routers” on page 96. For more information about creating policies, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

2. To apply the policy, include the `vrf-export` statement at the `[edit routing-instances routing-instance-name]` hierarchy level:

```
[edit routing-instances routing-instance-name]
vrf-export export-policy-name;
```

Apply Both the VRF Export and the BGP Export Policies

When you apply a VRF export policy as described in “Configure Export Policy for the PE Router's VRF Table” on page 91, routes from Layer 2 VPN (l2vpn) or Layer 3 VPN (vrf) routing instances are advertised to other PE routers based on this policy, while the BGP export policy is ignored.

If you configure the `vpn-apply-export` statement, both the VRF export and BGP group or neighbor export policies are applied (VRF first, then BGP) before routes are advertised in the vrf or l2vpn routing tables to other PE routers.

Configure the `vpn-apply-export` statement at the `[edit protocols bgp]`, `[edit protocols bgp group group-name]`, or `[edit protocols bgp group group-name neighbor neighbor]` hierarchy level:

```
[edit]
vpn-apply-export;
```

Configure a VRF Target

Before JUNOS 5.5, you needed to configure VRF import and export policies for each VPN routing instance on a PE router. These policies control redistribution of routes between the VRF table and BGP.

In the current JUNOS release, the `vrf-target` statement simplifies this configuration. Configuring a VRF target community using the `vrf-target` statement causes default VRF import and export policies to be generated which, respectively, accept and tag routes with the specified target community. You can still create more complex policies by explicitly configuring VRF import and export policies. Note that these policies would override the default policies generated when you configure the `vrf-target` statement.

If you do not configure the import and export options of the `vrf-target` statement, the specified community string is applied in both directions. The import and export keywords give you more flexibility, allowing you to specify a different community for each direction.

An example of how you might configure the `vrf-target` statement follows:

```
[edit routing-instances sample]
vrf-target target:69:102;
```

Note that the syntax for the VRF target community is not a name. You must specify it in the format `target:x.y`. A community name cannot be specified because this would also require you to configure the community members for that community using the `policy-options` statement. If you define the `policy-options` statements, then you can just configure VRF import and export policies as usual. The purpose of the `vrf-target` statement is to simplify the configuration by allowing you to configure most statements at the `[edit routing-instances]` hierarchy level.

To configure a VRF target, include the `vrf-target` statement at the `[edit routing-instances routing-instance-name]` hierarchy level:

```
[edit]
routing-instances {
  routing-instance-name {
    vrf-target community;
  }
}
```

To configure the `vrf-target` statement with the `export` and `import` options, include the following statements at the `[edit routing-instances routing-instance-name]` hierarchy level

```
[edit]
routing-instances {
  routing-instance-name {
    vrf-target {
      export community-name;
      import community-name;
    }
  }
}
```

Filter Traffic Based on the IP Header

The `vrf-table-label` statement makes it possible to map the inner label to a specific VRF and thus allow the examination of the encapsulated IP header at an egress VPN router. You might want to enable this functionality so you can do either of the following:

- Forward traffic on a PE-router-to-CE-device interface, in a shared medium, where the CE device is a Layer 2 switch without IP capabilities (for example, a metro Ethernet switch).

The first lookup is done on the VPN label to determine which VRF table to refer to, and the second lookup is done on the IP header to determine how to forward packets to the correct end hosts on the shared medium.

- Perform egress filtering at the egress PE router.

The first lookup on the VPN label is done to determine which VRF table to refer to, and the second lookup is done on the IP header to determine how to filter and forward packets. You can enable this functionality by configuring output filters on the VRF interfaces.



Note

The `vrf-table-label` statement is supported on the M-series platforms only. It is not supported on the T-series platforms.

When you use the `vrf-table-label` statement to configure a VRF table, a label-switched interface (LSI) logical interface label is created and mapped to the VRF. You perform this configuration at the `[edit routing-instances routing-instance-name]` hierarchy level.

Any routes configured in a VRF with the `vrf-table-label` statement are advertised with the LSI logical interface label allocated for the VRF. When packets for this VPN arrive on a core-facing interface, they are treated as if the enclosed IP packet arrived on the LSI interface and are then forwarded and filtered based on the correct table.

To filter traffic based on the IP header, include the `vrf-table-label` statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit routing-instances routing-instance-name]  
vrf-table-label;
```



Note

Do not use the `vrf-table-label` statement for source class usage/destination class usage (SCU/DCU) configurations. For information on SCU/DCU configuration, see the *JUNOS Internet Software Configuration Guide: Interfaces and Class of Service*.

Egress Filtering Options and Limitations

Egress filtering (which allows egress Layer 3 VPN PE routers to perform lookups on the VPN label and IP header at the same time) can be enabled by including the `vrf-table-label` statement at the [edit routing-instances *instance-name*] hierarchy level. However, this feature works only for non-channelized Point-to-Point Protocol/High-level Data Link Control (PPP/HDLC) core-facing SONET interfaces. Note that the `vrf-table-label` statement cannot be configured for the 10-port E1 Physical Interface Card (PIC) or for aggregated interfaces. There is no restriction on CE-router-to-PE-router interfaces.

You can also enable egress filtering by configuring a VPN tunnel (VT) interface on routers equipped with a Tunnel Services PIC. When you enable egress filtering this way, there is no restriction on the type of core-facing interface used. There is also no restriction on the type of CE-router-to-PE-router interface used.



Note

You cannot configure a VT interface and the `vrf-table-label` statement at the same time.

Configure a VPN Tunnel for VRF Table Lookup

You can configure a VPN tunnel to facilitate VRF table lookup based on MPLS labels. You might want to enable this functionality to forward traffic on a PE-router-to-CE-device interface in a shared medium, where the CE device is a Layer 2 switch without IP capabilities (for example, a metro Ethernet switch), or to perform egress filtering at the egress PE router.

For more information on VPN tunnels and VT interfaces, see the *JUNOS Internet Software Configuration Guide: Interfaces and Class of Service*.

Configure VPN Routing between the PE and CE Routers

For the PE router to distribute VPN-related routes to and from connected CE routers, you must configure routing within the VPN routing instance. You can configure a routing protocol—BGP, OSPF, or RIP—or you can configure static routing. For the connection to each CE router, you can configure only one type of routing.

This section describes how to do the following tasks:

- Configure BGP between the PE and CE Routers on page 96
- Configure OSPF between the PE and CE Routers on page 96
- Configure RIP between the PE and CE Routers on page 100
- Configure Static Routes between the PE and CE Routers on page 101
- Limit the Routes Accepted from a CE Router on page 101
- Configure IPv6 between the PE and CE Routers on page 102
- Configure EBGP or IBGP Multihop between PE and CE Routers on page 105

Configure BGP between the PE and CE Routers

To configure BGP as the routing protocol between the PE and the CE routers, include the `bgp` statement at the [edit routing-instances *routing-instance-name* protocols] hierarchy level:

```
[edit routing-instances routing-instance-name protocols]
bgp {
  group group-name {
    peer-as as-number;
    neighbor ip-address;
  }
}
```

Configure OSPF between the PE and CE Routers

You can configure OSPF to distribute VPN-related routes between PE and CE routers.

To configure OSPF as the routing protocol between a PE and CE router, include the `ospf` statement at the [edit routing-instances *routing-instance-name* protocols] hierarchy level:

```
[edit routing-instances routing-instance-name protocols]
ospf {
  area area {
    interface interface-name;
  }
}
```


Configure an OSPF Domain ID

For most OSPF configurations involving Layer 3 VPNs, you do not need to configure an OSPF domain ID. However, for a Layer 3 VPN connecting multiple OSPF domains, configuring OSPF domain IDs can help you control link-state advertisement (LSA) translation (for Type 3 and Type 5 LSAs) between the OSPF domains and back-door paths. The default OSPF domain ID is 0.0.0.0. Each VRF table in a PE router associated with an OSPF instance is configured with the same OSPF domain ID.

When a PE router receives a route, it redistributes and advertises the route either as a Type 3 LSA or as a Type 5 LSA, depending on the following:

- If the receiving PE router sees a Type 3 route with a matching domain ID, the route is redistributed and advertised as a Type 3 LSA.
- If the receiving PE router sees a Type 3 route without a domain ID (the extended attribute field of the route's BGP update does not include a domain ID), the route is redistributed and advertised as a Type 3 LSA.
- If the receiving PE router sees a Type 3 route with a non-matching domain ID, the route is redistributed and advertised as a Type 5 LSA.
- If the receiving PE router sees a Type 3 route with a domain ID but the receiving PE router does not have a domain ID configured, the route is redistributed and advertised as a Type 5 LSA.
- If the receiving PE router sees a Type 5 route, the route is redistributed and advertised as a Type 5 LSA, irrespective of the domain ID.

To configure an OSPF domain ID, include the domain-id statement at the [edit routing-instances *routing-instance-name* protocols ospf] hierarchy level:

```
[edit routing-instances routing-instance-name protocols]
ospf {
  domain-id domain ID;
}
```

You can set a VPN tag for the OSPF external routes generated by the PE router. This is used to prevent looping when a domain ID is used as an alternate route preference. By default, this tag is automatically calculated and needs no configuration. To configure the domain VPN tag for Type 5 LSAs, include the domain-vpn-tag *number* statement at the [edit routing-instances *routing-instance-name* protocols ospf] hierarchy level:

```
[edit routing-instances routing-instance-name protocols]
ospf {
  domain-vpn-tag number;
}
```

The range is 1 through 4,294,967,295. If you set VPN tags manually, you must set the same value for all PE routers in the VPN.

Hub-and-Spoke Layer 3 VPNs and OSPF Domain ID

The default behavior of an OSPF domain ID can cause the following problems for hub-and-spoke Layer 3 VPNs using OSPF between the PE and CE routers:

- PE routers set the down (DN) bit on all OSPF summary LSAs originating from area 0. PE routers are designated as area 0 by default because of the OSPF domain ID. When a PE router receives a summary LSA with the DN bit set, it drops the LSA. This is done to prevent routing loops.

For a hub-and-spoke Layer 3 VPN, when the hub PE router generates an OSPF summary LSA, it also sets the DN bit before sending it to the hub CE router. When the hub CE router sends the LSA back to the PE router, the PE router drops the LSA because the DN bit is set. Routes aggregated within the CE router are not affected by this problem.

- PE routers generating external LSAs learned from BGP updates set the `vpn-route-tag` field to a value derived from the PE router's AS number and an arbitrary tag. When a PE router receives an external LSA with a `vpn-route-tag` field that matches its own `vpn-route-tag` field, it drops the LSA. This is done to prevent routing loops.

For a hub-and-spoke Layer 3 VPN, an external LSA originated by a hub PE router is sent to the hub CE router, which then sends it back to the same PE router. Because the `vpn-route-tag` field matches the PE router's `vpn-route-tag` field, the LSA is dropped. Routes aggregated within the CE router are not affected by this problem.

For hub-and-spoke Layer 3 VPNs using OSPF between the PE and CE routers to work, you need to configure the following on the hub PE router:

- Configure the `disable` statement at the `[edit routing-instances routing-instance-name protocols ospf]` hierarchy level on the routing instance for the hub CE router. This removes area 0 from the PE router, allowing the PE router to forward LSAs without setting the DN bit. When an LSA comes back from the hub CE router, the PE router can install it because the DN bit is not set.
- Configure 0 for the `vpn-route-tag` statement at the `[edit routing-instances routing-instance-name protocols ospf]` hierarchy level on the routing instance for the spoke CE router. This removes any VPN route tags that are set on the external LSAs, preventing a VPN route tag match and allowing the PE router to install the LSA.

Compatibility with JUNOS Releases before 5.3

For JUNOS release 5.3, the format for `domain-id`, an extended community type defined in the BGP extended community attribute field, was modified to comply with the Internet Engineering Task Force (IETF) draft `draft-rosen-vpns-ospf-bgp-mpls` (available at <http://www.ietf.org/>). JUNOS releases prior to 5.3 continue to use the previously supported vendor-specific formats.

The OSPF domain ID format is incompatible between JUNOS 5.3 or later and JUNOS 5.2 or earlier. For OSPF domain IDs to function properly between a PE router running JUNOS 5.3 or later and a PE router running JUNOS 5.2 or earlier, you need to define the extended community type for the BGP extended community attribute field as domain-id-vendor (instead of as domain-id). This is part of the policy-options configuration for the OSPF domain ID configured at the [edit policy-options community vrf_export_attributes members] hierarchy level:

```
[edit policy-options community vrf_export_attributes members]
domain-id-vendor:ip-address
```

You also need to configure the route-type-community statement with the vendor option at the [edit routing-instances *routing-instance-name* protocols ospf] hierarchy level:

```
[edit routing-instances routing-instance-name protocols]
ospf {
  route-type-community vendor;
}
```

The default value for the route-type-community statement is iana.

Example Configurations for Compatibility with JUNOS Releases before 5.3

The following example shows a configuration of the policy options for a PE router. The PE router has an OSPF domain ID configured.

It needs to be compatible with a router running a pre-5.3 version of JUNOS software. As a part of the community statement configuration, specify domain-id-vendor for the attribute that assigns the domain ID instead of domain-id:

```
[edit]
policy-options {
  policy-statement vrf_import_routes {
    term a {
      from {
        protocol bgp;
        community vrf_import_attributes;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement vrf_export_routes {
    term a {
      from protocol ospf;
      then {
        community add vrf_export_attributes;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community vrf_export_attributes members [ target:10.19.2.0:5 domain-id-vendor:1.2.3.4:0 ];
  community vrf_import_attributes members target:10.19.1.0:5;
}
```

The following example shows a configuration for a routing instance on a PE router. The PE router has an OSPF domain ID configured. It needs to be compatible with a router running an earlier version of JUNOS software. The configuration includes the route-type-community statement with the vendor option. This is so the PE router receiving the route knows how to parse the incoming BGP attribute field containing the domain ID.

The example configuration follows:

```
[edit]
routing-instances {
  CE_A {
    instance-type vrf;
    interface fe-1/0/0.0;
    route-distinguisher 10.255.25.270:1;
    vrf-import vrf_import_routes;
    vrf-export vrf_export_routes;
    protocols {
      ospf {
        route-type-community vendor;
        domain-id 1.2.3.4;
        export vrf_import_routes;
        area 0.0.0.0 {
          interface fe-1/0/0.0;
        }
      }
    }
  }
}
```

Configure RIP between the PE and CE Routers

For a Layer 3 VPN, you can configure RIP on the PE router to learn the routes of the CE router or to propagate the routes of the PE router to the CE router. RIP routes learned from neighbors configured at any [edit routing-instances] hierarchy level are added to the routing instance's inet table (*instance_name.inet.0*).

To configure RIP as the routing protocol between the PE and the CE router, include the rip statement at the [edit routing-instances *routing-instance-name* protocols] hierarchy level:

```
[edit routing-instances routing-instance-name protocols]
rip {
  group group-name {
    neighbor interface-name;
  }
}
```

To install routes learned from a RIP routing instance to multiple routing tables, configure the rib-group statement at the [edit protocols rip] hierarchy level or at the [edit routing-instances *routing-instance-name* protocols rip] hierarchy level:

```
[edit protocols rip]
rib-group inet group-name;
group group-name {
  neighbor interface-name;
}
```

To configure a routing table group, configure the rib-group statement at the [edit routing-options] hierarchy level.

To add a routing table to a routing table group, you need to configure the `import-rib` statement at the `[edit routing-options rib-groups group-name]` hierarchy level. The first routing table name specified under the `import-rib` statement must be the name of the routing table you are configuring. See the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols* for more information about how to configure routing tables and routing table groups.

Configure the `import-rib` statement at the `[edit routing-options rib-groups group-name]` hierarchy level as follows:

```
[edit routing-options rib-groups group-name]
import-rib [group-name]
```

Configure Static Routes between the PE and CE Routers

To configure a static route between the PE and the CE routers, include the `routing-options static` statement at the `[edit routing-instances routing-instance-name routing-options]` hierarchy level:

```
[edit routing-instances routing-instance-name routing-options]
static {
  route destination-prefix {
    next-hop;
    static-options;
  }
}
```

For more information about configuring routing protocols and static routes, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

Limit the Routes Accepted from a CE Router

A route limit sets an upper limit for the number of prefixes installed into routing tables. You can use route limits to curtail the number of routes received from a CE router in a VPN. A route limit applies only to dynamic routing protocols, and is not applicable to static or interface routes.

To limit the number of routes accepted by a PE router from a CE router, include the `maximum-routes` statement at the `[edit routing-instances routing-instance-name routing-options]` hierarchy level:

```
[edit routing-instances routing-instance-name routing-options]
maximum-routes route-limit <log-only | threshold value>;
```

There are two modes for route limits: advisory (set with the `log-only` option) and mandatory. An advisory limit triggers only warnings. The log messages are rate-limited to once every 30 seconds. A mandatory limit, in addition to triggering a warning message, rejects any additional routes after the threshold is reached. The threshold value is a percentage of the route limit at which warning messages are logged.



Note

Setting a route limit might result in unpredictable dynamic routing protocol behavior.

Configure IPv6 between the PE and CE Routers

You can configure IPv6 between the PE and CE routers of a Layer 3 VPN. The PE router must have the PE router to PE router BGP session configured with the family inet6-vpn statement. The CE router must be capable of receiving IPv6 traffic. You can configure BGP or static routes between the PE and CE routers.

To configure IPv6 VPNS between the PE routers, complete the following steps:

- Configure IPv6 on the PE Router on page 102
- Configure BGP or Static Routes on the PE Router on page 102
- Configure IPv6 on the Interfaces on page 104

Configure IPv6 on the PE Router

To configure IPv6 between the PE and CE routers, include the following statements at the [edit protocols bgp group *group-name*] hierarchy level on the PE router:

```
[edit protocols bgp group group-name]
family inet6-vpn {
  (unicast | multicast | any) {
    prefix-limit maximum prefix-limit;
    rib-group rib-group-name;
  }
}
```

Configure the ipv6-tunneling statement at the [edit protocols mpls] hierarchy level:

```
[edit protocols mpls]
ipv6-tunneling;
```

Configure BGP or Static Routes on the PE Router

You must configure either BGP or static routes for the connection between the PE and CE routers in the Layer 3 VPN. You can configure BGP to handle just IPv6 routes or both IPv4 and IPv6 routes.

For more information about IPv6, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

The following sections describe how to configure BGP and static routes:

- Configure BGP on the PE Router to Handle IPv6 Routes on page 103
- Configure BGP on the PE Router to Handle IPv4 and IPv6 Routes on page 103
- Configure Static Routes on the PE Router on page 104

Configure BGP on the PE Router to Handle IPv6 Routes

Configure BGP in the Layer 3 VPN routing instance to handle IPv6 routes at the [edit routing-instances *routing-instance-name* protocols bgp] hierarchy level:

```
[edit]
routing-instances routing-instance-name {
  protocols {
    bgp {
      group group-name {
        local-address IPv6-address;
        family inet6 {
          unicast;
        }
        peer-as as-number;
        neighbor IPv6-address;
      }
    }
  }
}
```

Configure BGP on the PE Router to Handle IPv4 and IPv6 Routes

Configure BGP in the Layer 3 VPN routing instance to handle both IPv4 and IPv6 routes at the [edit routing-instances *routing-instance-name* protocols bgp] hierarchy level:

```
[edit]
routing-instances routing-instance-name {
  protocols {
    bgp {
      group group-name {
        local-address IPv4-address;
        family inet {
          unicast;
        }
        family inet6 {
          unicast;
        }
        peer-as as-number;
        neighbor address;
      }
    }
  }
}
```

Configure Static Routes on the PE Router

Configure a static route to the CE router in the Layer 3 VPN routing instance at the [edit routing-instances *routing-instance-name* routing-options rib *routing-table-group-name*] hierarchy level:

```
[edit routing-instances routing-instance-name]
routing-options {
  rib routing-table-group-name.inet6.0 {
    static {
      defaults {
        static-options;
      }
    }
  }
}
```

Configure IPv6 on the Interfaces

You need to configure IPv6 on the PE router interfaces to the CE routers and on the CE router interfaces to the PE routers.

To configure the interface to handle IPv6 routes, include the family inet6 statement under the [edit interfaces *interface-name* unit *unit-number*] hierarchy level:

```
[edit]
interfaces {
  interface-name {
    unit unit-number {
      family inet6 {
        address IPv6-address;
      }
    }
  }
}
```

If you have configured the Layer 3 VPN to handle both IPv4 and IPv6 routes, you need to configure the interface to handle both IPv4 and IPv6 routes:

```
[edit]
interfaces {
  interface-name {
    unit unit-number {
      family inet {
        address IPv4-address;
      }
      family inet6 {
        address IPv6-address;
      }
    }
  }
}
```


Configure EBGp or IBGP Multihop between PE and CE Routers

You can configure an external BGP (EBGP) or internal BGP (IBGP) multihop session between the PE and CE routers of a Layer 3 VPN. This allows you to have one or more routers between the PE and CE routers. Using IBGP between PE and CE routers does not require the configuration of any additional statements. However, using EBGP between the PE and CE routers requires the configuration of the multihop statement.

To configure an external BGP multihop session for the connection between the PE and CE routers, include the multihop statement at the [edit routing-instances *routing-instance-name* protocols bgp], [edit routing-instances *routing-instance-name* protocols bgp group *group-name*], or [edit routing-instances *routing-instance-name* protocols bgp group *group-name* neighbor *address*] hierarchy level:

```
multihop <ttl-value>;
```

Configure VPN Graceful Restart

To enable VPN graceful restart, configure the graceful-restart statement at the [edit routing-options] hierarchy level on the PE router:

```
[edit routing-options]
graceful-restart {...}
```

Also include the graceful-restart statement at the [edit routing-instance *routing-instance-name* routing-options] hierarchy level on the PE router:

```
[edit routing-instance routing-instance-name routing-options]
graceful-restart {...}
```

Configure Multicast over Layer 3 VPNs

You can configure a Layer 3 VPN to support multicast traffic using the Protocol Independent Multicast (PIM) routing protocol. To support multicast, you need to configure PIM on routers within the VPN and within the service provider's network.

Each PE router configured to run multicast over Layer 3 VPNs must have a Tunnel PIC. A Tunnel PIC is also required on the provider routers that act as rendezvous points (RPs). Tunnel PICs are also needed on all the CE routers acting as designated routers (first-hop/last-hop routers) or as RPs, just as they are in non-VPN PIM environments.

Configure the master PIM instance at the [edit protocols pim] hierarchy level on the CE and PE routers. You also need to configure a PIM instance for the Layer 3 VPN at the [edit routing-instances *routing-instance-name* protocols pim] hierarchy level on the PE router. This creates a PIM instance for the indicated routing instance.

For information about how to configure PIM, see the *JUNOS Internet Software Configuration Guide: Multicast*.

The vpn-group-address statement is unique to a Layer 3 VPN PIM configuration. You use this statement to configure the group address for the VPN in the service provider's network. This address should be unique for each VPN. It ensures that multicast traffic is transmitted only to the specified VPN.

Configure the `vpn-group-address` statement at the `[edit routing-instances routing-instance-name protocols pim]` hierarchy level:

```
[edit routing-instances routing-instance-name protocols]
pim {
  vpn-group-address address;
}
```

The rest of the Layer 3 VPN configuration for multicast is conventional and is described in other sections of this manual. Most of the specific configuration tasks needed to activate multicast in a VPN environment involve PIM. For more information about how to configure PIM and multicast in JUNOS, including an example of how to configure multicast over Layer 3 VPNs, see the *JUNOS Internet Software Configuration Guide: Multicast*.

Configure a GRE Tunnel Interface for Layer 3 VPNs

JUNOS software allows you to configure a generic routing encapsulation (GRE) tunnel between the PE and CE routers for a Layer 3 VPN. The GRE tunnel can have one or more hops.

For more information about how to configure tunnel interfaces, see the *JUNOS Internet Software Configuration Guide: Interfaces and Class of Service*.

To configure a GRE tunnel between the PE and CE routers for a Layer 3 VPN, complete the procedures in the following sections:

- Configure the GRE Tunnel Interface on the PE Router on page 106
- Configure the GRE Tunnel Interface on the CE Router on page 107

Configure the GRE Tunnel Interface on the PE Router

Configure the GRE tunnel interface on the PE router as follows:

```
[edit]
interfaces {
  interface-name {
    unit 0 {
      tunnel {
        source address;
        destination address;
      }
      family inet {
        address address;
      }
    }
  }
}
```

By default, the tunnel destination address is assumed to be in the default Internet routing table, `inet.0`. If the tunnel destination address is not in `inet.0`, you need to specify which routing table to search for the tunnel destination address by configuring the `routing-instance` statement. This is the case if the tunnel encapsulating interface is also configured under the routing instance.

Configure the GRE tunnel interface on the PE router and specify the name of the routing instance:

```
[edit]
interfaces {
  interface-name {
    unit 0 {
      tunnel {
        source address;
        destination address;
        routing-instance {
          destination routing-instance-name;
        }
        family inet {
          address address;
        }
      }
    }
  }
}
```

To complete the GRE tunnel interface configuration, you need to configure the GRE interface at the [edit routing-instances *routing-instance-name*] hierarchy level under the appropriate routing-instance:

```
[edit]
routing-instances {
  routing-instance-name {
    interface interface-name;
  }
}
```

Configure the GRE Tunnel Interface on the CE Router

Configure the GRE tunnel interface on the CE router as follows:

```
[edit]
interfaces {
  interface-name {
    unit 0 {
      tunnel {
        source address;
        destination address;
      }
      family inet {
        address address;
      }
    }
  }
}
```

Configure an ES Tunnel Interface for Layer 3 VPNs

An ES tunnel interface allows you to configure an IP Security (IPSec) tunnel between the PE and CE routers of a Layer 3 VPN. The IPSec tunnel can include one or more hops.

To configure an ES tunnel interface between the PE and CE routers of a Layer 3 VPN, complete the procedures in the following sections:

- Configure the ES Tunnel Interface on the PE Router on page 108
- Configure the ES Tunnel Interface on the CE Router on page 109

Configure the ES Tunnel Interface on the PE Router

Configure the ES tunnel interface on the PE router as follows:

```
[edit]
interfaces {
  interface-name {
    unit 0 {
      tunnel {
        source address;
        destination address;
      }
      family inet {
        address address;
        ipsec-sa security-association-name;
      }
    }
  }
}
```

By default, the tunnel destination address is assumed to be in the default Internet routing table, inet.0. For IPSec tunnels using manual security association (SA), if the tunnel destination address is not in the default inet.0 routing table, you need to specify which routing table to search for the tunnel destination address by configuring the routing-instance statement. This is the case if the tunnel encapsulating interface is also configured under the routing instance.

```
[edit]
interfaces {
  interface-name {
    unit 0 {
      tunnel {
        source address;
        destination address;
        routing-instance {
          destination routing-instance-name;
        }
      }
      family inet {
        address address;
        ipsec-sa security-association-name;
      }
      family mpls;
    }
  }
}
```



For IPSec tunnels using dynamic SA, the tunnel destination address must be in the default Internet routing table, inet.0.

You also need to configure the ES interface at the [edit routing-instances *routing-instance-name*] hierarchy level for the appropriate routing instance:

```
[edit]
routing-instances {
  routing-instance-name {
    interface interface-name;
  }
}
```

Configure the ES Tunnel Interface on the CE Router

Configure the ES tunnel interface on the CE router as follows:

```
[edit]
interfaces {
  interface-name {
    unit 0 {
      tunnel {
        source address;
        destination address;
      }
      family inet {
        address address;
        ipsec-sa security-association-name;
      }
    }
  }
}
```

For more information about how to configure tunnel interfaces, see the *JUNOS Internet Software Configuration Guide: Interfaces and Class of Service*.

For more information about how to configure IPSec interfaces, see the *JUNOS Internet Software Configuration Guide: Getting Started*.

Configure IPsec between PE Routers Instead of MPLS

A conventional Layer 3 BGP/MPLS VPN requires the configuration of MPLS LSPs between the PE routers. When a PE router receives a packet from a CE router, it performs a lookup in a specific VRF table for the IP destination address and obtains a corresponding MPLS label stack. The label stack is used to forward the packet to the egress PE router, where the bottom label is removed and the packet is forwarded to the specified CE router.

You can provide Layer 3 BGP/MPLS VPN service without an MPLS backbone. Instead of configuring MPLS LSPs between the PE routers, you configure GRE and IPsec tunnels between the PE routers. The MPLS information for the VPN (the VPN label) is encapsulated within an IP header and an IPsec header. The source address of the IP header is the address of the ingress PE router. The destination address has the BGP next hop, the address of the egress PE router.



The IPsec tunnel requires the use of an ES PIC. The GRE tunnel requires the use of a Tunnel Services PIC.

To configure IPsec between PE routers, complete the following:

1. Configure an IPsec tunnel between the PE routers. The source address is that of the ingress PE router, and the destination address is that of the egress PE router:

```
[edit interfaces]
es-interface-name {
  unit unit-number {
    tunnel {
      source source-address;
      destination destination-address;
    }
    family inet {
      ipsec-sa sa-esp-dynamic;
      address address;
    }
    family mpls;
  }
}
```

2. Configure IPsec on the PE router. For information about how to configure IPsec, see the *JUNOS Internet Software Configuration Guide: Getting Started*.

3. Configure a GRE tunnel between the PE routers. Again, the source address is that of the ingress PE router, and the destination address is that of the egress PE router:

```
[edit interfaces]
gr-interface-name {
  unit unit-number {
    family inet {
      address address;
    }
    family mpls;
    tunnel {
      source source-address;
      destination destination-address;
    }
  }
}
```

4. Configure BGP between the PE routers:

```
[edit protocols]
bgp {
  group pe {
    type internal;
    local-address local-address;
    family inet {
      unicast;
    }
    family inet-vpn {
      unicast;
    }
    peer-as as-number;
    neighbor address;
  }
}
```

5. Configure the routing instance:

```
[edit]
routing-instances {
  routing-instance-name {
    instance-type vrf;
    interface interface-name;
    route-distinguisher address;
    vrf-import import-policy-name;
    vrf-export export-policy-name;
    protocols {
      bgp {
        group routing-instance-name {
          type external;
          peer-as as-number;
          as-override;
          neighbor address;
        }
      }
    }
  }
}
```

6. Configure the policy options:

```
[edit]
policy-options {
  policy-statement import-policy-name {
    term 1 {
      from {
        protocol bgp;
        community community-name;
      }
      then accept;
    }
    term 2 {
      then reject;
    }
  }
  policy-statement export-policy-name {
    term 1 {
      from protocol [ bgp direct ];
      then {
        community add community-name;
        accept;
      }
    }
    term 2 {
      then reject;
    }
  }
  community community-name members target:target;
}
```

Configure Packet Forwarding for VPNs

The procedure for configuring packet forwarding in Layer 3 VPNs is identical to the one described for Layer 2 VPNs in “Configure Packet Forwarding for VPNs” on page 26.

Chapter 8

Layer 3 VPN Configuration Troubleshooting Guidelines

This chapter discusses the following strategies and tools for troubleshooting Layer 3 virtual private network (VPN) configurations:

- Diagnose Common Problems on page 113
- Use the ping and traceroute Commands to Troubleshoot Layer 3 VPN Topologies on page 117
- Indirect Next-hop Address Space and Route Reflectors on page 125

Diagnose Common Problems

When problems arise in a Layer 3 VPN configuration, the best way to troubleshoot is to start at one end of the VPN (the local customer edge [CE] router) and follow the routes to the other end of the VPN (the remote CE router). The following troubleshooting steps should help you diagnose common problems:

1. If you configured a routing protocol between the local provider edge (PE) and CE routers, check that the peering and adjacency are fully operational. When you do this, be sure to specify the name of the routing instance. For example, to check Open Shortest Path First (OSPF) adjacencies, enter the command `show ospf neighbor instance routing-instance-name` on the PE router.

If the peering and adjacency are not fully operational, check the routing protocol configuration on the CE router and check the routing protocol configuration for the associated VPN routing instance on the PE router.

2. Check that the local CE and PE routers can ping each other.

To check that the local CE router can ping the VPN interface on the local PE router, use a ping command in the following format, specifying the IP address or name of the PE router:

```
ping (ip-address | host-name)
```

To check that the local PE router can ping the CE router, use a ping command in the following format, specifying the IP address or name of the CE router, the name of the interface used for the VPN, and the source IP address (the local address) in outgoing ECHO_REQUEST packets:

```
ping ip-address vpn-interface interface local echo-address
```

Often, the peering or adjacency between the local CE and local PE routers needs to come up before a ping command is successful. To check that a link is operational in a lab setting, remove the interface from the VRF by deleting the interface statement from the [edit routing-instance *routing-instance-name*] hierarchy level and recommitting the configuration. Doing this removes the interface from the VPN. Then try the ping command again. If the command is successful, configure the interface back into the VPN and check the routing protocol configuration on the local CE and PE routers again.

3. On the local PE router, check that the routes from the local CE router are in the VPN routing and forwarding (VRF) table (*routing-instance-name.inet.0*):

```
show route table routing-instance-name.inet.0 [detail]
```

The following example shows the routing table entries. Here, the loopback address of the CE router is 10.255.14.155/32 and the routing protocol between the PE and CE routers is Border Gateway Protocol (BGP). The entry looks like any ordinary BGP announcement.

```
10.255.14.155/32 (1 entry, 1 announced)
  *BGP    Preference: 170/-101
          Nexthop: 192.168.197.141 via fe-1/0/0.0, selected
          State: <Active Ext>
          Peer AS:      1
          Age: 45:46
          Task: BGP_1.192.168.197.141+179
          Announcement bits (2): 0-BGP.0.0.0.0+179 1-KRT
          AS path: 1 I
          Localpref: 100
          Router ID: 10.255.14.155
```

If the routes from the local CE router are not present in the VRF routing table, check that the CE router is advertising routes to the PE router. If static routing is used between the CE and PE routers, make sure the proper static routes are configured.

4. On a remote PE router, check that the routes from the local CE router are present in the bgp.l3vpn.0 routing table:

```
show route table bgp.l3vpn.0 extensive
```

The following example shows the routing table entries.

```
10.255.14.175:3:10.255.14.155/32 (1 entry, 0 announced)
  *BGP    Preference: 170/-101
          Route Distinguisher: 10.255.14.175:3
          Source: 10.255.14.175
          Nexthop: 192.168.192.1 via fe-1/1/2.0, selected
          label-switched-path vpn07-vpn05
          Push 100004, Push 100005(top)
          State: <Active Int Ext>
          Local AS:      69 Peer AS:      69
          Age: 15:27      Metric2: 338
          Task: BGP_69.10.255.14.175+179
          AS path: 1 I
          Communities: target:69:100
          BGP next hop: 10.255.14.175
          Localpref: 100
          Router ID: 10.255.14.175
          Secondary tables: VPN-A.inet.0
```

The output of the `show route table bgp.l3vpn.0` extensive command contains the following information specific to the VPN:

- In the prefix name (the first line of the output), the route distinguisher is added to the route prefix of the local CE router. Because the route distinguisher is unique within the Internet, the concatenation of the route distinguisher and IP prefix provides unique VPN-Internet Protocol Version 4 (IPv4) routing entries.
- The Route Distinguisher field lists the route distinguisher separately from the VPN-IPv4 address.
- The label-switched-path field shows the name of the label-switched path (LSP) used to carry the VPN traffic.
- The Push field shows both labels being carried in the VPN-IPv4 packet. The first label is the inner label, which is the VPN label that was assigned by the PE router. The second label is the outer label, which is a Resource Reservation Protocol (RSVP) label.
- The Communities field lists the target community.
- The Secondary tables field lists other routing tables on this router into which this route has been installed.

If routes from the local CE router are not present in the `bgp.l3vpn.0` routing table on the remote PE router, do the following:

- Check the VRF import filter on the remote PE router, which is configured in the `vrf-import` statement. (On the local PE router, you check the VRF export filter, which is configured with the `vrf-export` statement.)
- Check that there is an operational LSP or a Label Distribution Protocol (LDP) path between the PE routers. To do this, check that the internal Border Gateway Protocol (IBGP) next-hop addresses are in the `inet.3` table.
- Check that the IBGP session between the PE routers is established and configured properly.
- Check for “hidden” routes, which usually means that routes were not labeled properly. To do this, use the `show route table bgp.l3vpn.0 hidden` command.
- Check that the inner label matches the inner VPN label that is assigned by the local PE router. To do this, use the `show route table mpls` command.

The following example shows the output of this command on the remote PE router. Here, the inner label is 100004.

```
...
Push 100004, Push 10005 (top)
```

The following example shows the output of this command on the local PE router, which shows that the inner label of 100004 matches the inner label on the remote PE router:

```
...
100004                *[VPN/7] 06:56:25, metric 1
> to 192.168.197.141 via fe-1/0/0.0, Pop
```

5. On the remote PE router, check that the routes from the local CE router are present in the VRF table (*routing-instance-name.inet.0*):

```
show route table routing-instance-name.inet.0 [detail]
```

The following example shows the routing table entries:

```
10.255.14.155/32 (1 entry, 1 announced)
  *BGP      Preference: 170/-101
            Route Distinguisher: 10.255.14.175:3
            Source: 10.255.14.175
            Nexthop: 192.168.192.1 via fe-1/1/2.0, selected
            label-switched-path vpn07-vpn05
            Push 100004, Push 100005(top)
            State: <Secondary Active Int Ext>
            Local AS: 69 Peer AS: 69
            Age: 1:16:22 Metric2: 338
            Task: BGP_69.10.255.14.175+179
            Announcement bits (2): 1-KRT 2-VPN-A-RIP
            AS path: 1 I
            Communities: target:69:100
            BGP next hop: 10.255.14.175
            Localpref: 100
            Router ID: 10.255.14.175
            Primary Routing Table bgp.l3vpn.0
```

In this routing table, the route distinguisher is no longer prepended to the prefix. The last line, Primary Routing Table, lists the table from which this route was learned.

If the routes are not present in this routing table, but were present in Step 4, the routes might have not passed the VRF import policy on the remote PE router.

If a VPN-IPv4 route matches no vrf-import policy, the route does not show up in the bgp.l3vpn table at all and hence is not present in the VRF table. If this occurs, it might indicate that on the PE router, you have configured another vrf-import statement on another VPN (with a common target), and the routes show up in the bgp.l3vpn.0 table, but are imported into the wrong VPN.

6. On the remote CE router, check that the routes from the local CE router are present in the routing table (inet.0):

```
show route
```

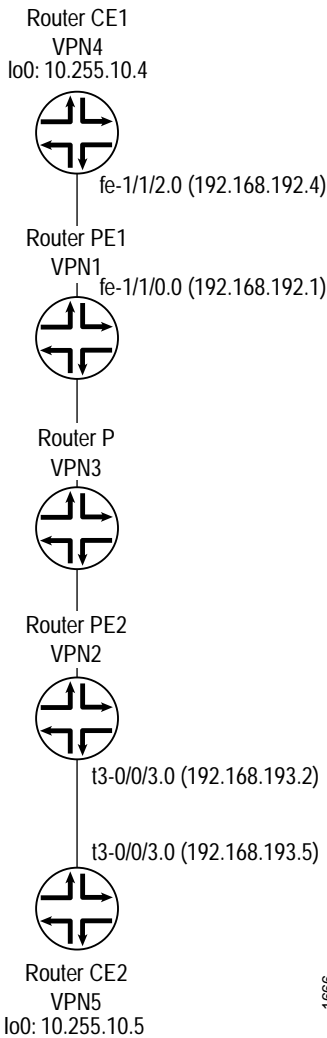
If the routes are not present, check the routing protocol configuration between the remote PE and CE routers, and make sure that peers and adjacencies (or static routes) between the PE and CE routers are correct.

7. If, in Steps 1 through 6, you have determined that routes originated from the local CE router are correct, check the routes originated from the remote CE router by repeating Steps 1 through 6.

Use the ping and traceroute Commands to Troubleshoot Layer 3 VPN Topologies

This section provides examples of how to use the ping command to check the accessibility of various routers in a VPN topology, and how to use the traceroute command to check the path that packets travel between the VPN routers. The topology shown in Figure 15 illustrates these commands.

Figure 15: Layer 3 VPN Topology for ping and traceroute Command Examples



Ping One CE Router from the Other

You can ping one CE router from the other by specifying the other CE router's loopback address as the IP address in the ping command. This ping command succeeds if the loopback addresses have been announced by the CE routers to their directly connected PE routers. The success of these ping commands also means that Router CE1 can ping any network devices beyond Router CE2, and vice versa. See Figure 15 for the topology referenced in these examples.

Ping Router CE2 (VPN5) from Router CE1 (VPN4):

```
user@vpn4> ping 10.255.10.5 local 10.255.10.4 count 3
PING 10.255.10.5 (10.255.10.5): 56 data bytes
64 bytes from 10.255.10.5: icmp_seq=0 ttl=253 time=1.086 ms
64 bytes from 10.255.10.5: icmp_seq=1 ttl=253 time=0.998 ms
64 bytes from 10.255.10.5: icmp_seq=2 ttl=253 time=1.140 ms

--- 10.255.10.5 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.998/1.075/1.140/0.059 ms
```

To determine the path from Router CE1's loopback interface to Router CE2's loopback interface, use the following traceroute command:

```
user@vpn4> traceroute 10.255.10.5 source 10.255.10.4
traceroute to 10.255.10.5 (10.255.10.5) from 10.255.10.4, 30 hops max, 40 byte
packets
 1  vpn1-fe-110.isp-core.net (192.168.192.1)  0.680 ms  0.491 ms  0.456 ms
 2  vpn2-t3-001.isp-core.net (192.168.192.110) 0.857 ms  0.766 ms  0.754 ms
    MPLS Label=100005 CoS=0 TTL=1 S=1
 3  vpn5.isp-core.net (10.255.10.5) 0.825 ms  0.886 ms  0.732 ms
```

Ping Router CE1 (VPN4) from Router CE2 (VPN5):

```
user@vpn5> ping 10.255.10.4 local 10.255.10.5 count 3
PING 10.255.10.4 (10.255.10.4): 56 data bytes
64 bytes from 10.255.10.4: icmp_seq=0 ttl=253 time=1.042 ms
64 bytes from 10.255.10.4: icmp_seq=1 ttl=253 time=0.998 ms
64 bytes from 10.255.10.4: icmp_seq=2 ttl=253 time=0.954 ms

--- 10.255.10.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.954/0.998/1.042/0.036 ms
```

To determine the path from Router CE2 to Router CE1, use the following traceroute command:

```
user@vpn5> traceroute 10.255.10.4 source 10.255.10.5
traceroute to 10.255.10.4 (10.255.10.4) from 10.255.10.5, 30 hops max, 40 byte
packets
 1  vpn-08-t3-003.isp-core.net (192.168.193.2) 0.686 ms  0.519 ms  0.548 ms
 2  vpn1-so-100.isp-core.net (192.168.192.100) 0.918 ms  0.869 ms  0.859 ms
    MPLS Label=100021 CoS=0 TTL=1 S=1
 3  vpn4.isp-core.net (10.255.10.4) 0.878 ms  0.760 ms  0.739 ms
```

Ping the Remote PE and CE Routers from the Local CE Router

From the local CE router, you can ping the VPN interfaces on the remote PE and CE routers, which are point-to-point interfaces. See Figure 15 for the topology referenced in these examples.

Ping Router CE2 (VPN5) from Router CE1 (VPN4):

```
user@vpn4> ping 192.168.193.5 local 10.255.10.4 count 3
PING 192.168.193.5 (192.168.193.5): 56 data bytes
64 bytes from 192.168.193.5: icmp_seq=0 ttl=253 time=1.040 ms
64 bytes from 192.168.193.5: icmp_seq=1 ttl=253 time=0.891 ms
64 bytes from 192.168.193.5: icmp_seq=2 ttl=253 time=0.944 ms

--- 192.168.193.5 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.891/0.958/1.040/0.062 ms
```

To determine the path from Router CE1's loopback interface to Router CE2's directly connected interface, use the following traceroute command:

```
serpil@vpn4> traceroute 192.168.193.5 source 10.255.10.4
traceroute to 192.168.193.5 (192.168.193.5) from 10.255.10.4, 30 hops max, 40
byte packets
 1  vpn1-fe-110.isp-core.net (192.168.192.1)  0.669 ms  0.508 ms  0.457 ms
 2  vpn2-t3-001.isp-core.net (192.168.192.110) 0.851 ms  0.769 ms  0.750 ms
    MPLS Label=100000 CoS=0 TTL=1 S=1
 3  vpn5-t3-003.isp-core.net (192.168.193.5)  0.829 ms  0.838 ms  0.731 ms
```

Ping Router PE2 (VPN2) from Router CE1 (VPN4). In this case, packets that originate at Router CE1 go to Router PE2, then to Router CE2, and back to Router PE2 before Router PE2 can respond to Internet Control Message Protocol (ICMP) requests. You can verify this using the traceroute command.

```
user@vpn4> ping 192.168.193.2 local 10.255.10.4 count 3
PING 192.168.193.2 (192.168.193.2): 56 data bytes
64 bytes from 192.168.193.2: icmp_seq=0 ttl=254 time=1.080 ms
64 bytes from 192.168.193.2: icmp_seq=1 ttl=254 time=0.967 ms
64 bytes from 192.168.193.2: icmp_seq=2 ttl=254 time=0.983 ms

--- 192.168.193.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.967/1.010/1.080/0.050 ms
```

To determine the path from Router CE1 to Router PE2, use the following traceroute command:

```
user@vpn4> traceroute 192.168.193.2 source 10.255.10.4
traceroute to 192.168.193.2 (192.168.193.2) from 10.255.10.4, 30 hops max, 40
byte packets
 1  vpn1-fe-110.isp-core.net (192.168.192.1)  0.690 ms  0.490 ms  0.458 ms
 2  vpn2-t3-003.isp-core.net (192.168.193.2) 0.846 ms  0.768 ms  0.749 ms
    MPLS Label=100000 CoS=0 TTL=1 S=1
 3  vpn5-t3-003.isp-core.net (192.168.193.5)  0.643 ms  0.703 ms  0.600 ms
 4  vpn-08-t3-003.isp-core.net (192.168.193.2) 0.810 ms  0.739 ms  0.729 ms
```

You cannot ping one CE router from the other if the VPN interface is a multi-access interface, such as the fe-1/1/2.0 interface on Router CE1. To ping Router CE1 from Router CE2, you must configure a static route on Router PE1 to the VPN interface of Router CE1 that has a next-hop pointing to Router CE1 (at the [edit routing-instance *routing-instance-name*] hierarchy level), and this route must be announced from Router PE1 to Router PE2. The following configuration portions illustrate this configuration:

```
[edit]
routing-instances {
  direct-multipoint {
    instance-type vrf;
    interface fe-1/1/0.0;
    route-distinguisher 69:1;
    vrf-import direct-import;
    vrf-export direct-export;
    routing-options {
      static {
        route 192.168.192.4/32 next-hop 192.168.192.4;
      }
    }
  }
  protocols {
    bgp {
      group to-vpn4 {
        peer-as 1;
        neighbor 192.168.192.4;
      }
    }
  }
}
policy-options {
  policy-statement direct-export {
    term a {
      from protocol bgp;
      then {
        community add direct-comm;
        accept;
      }
    }
    term b {
      from {
        protocol static;
        route-filter 192.168.192.4/32 exact;
      }
      then {
        community add direct-comm;
        accept;
      }
    }
    term d {
      then reject;
    }
  }
}
```


Now you can ping Router CE1 from Router CE2:

```
user@vpn5> ping 192.168.192.4 local 10.255.10.5 count 3
PING 192.168.192.4 (192.168.192.4): 56 data bytes
64 bytes from 192.168.192.4: icmp_seq=0 ttl=253 time=1.092 ms
64 bytes from 192.168.192.4: icmp_seq=1 ttl=253 time=1.019 ms
64 bytes from 192.168.192.4: icmp_seq=2 ttl=253 time=1.031 ms

--- 192.168.192.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.019/1.047/1.092/0.032 ms
```

To determine the path between these two interfaces, use the following traceroute command:

```
user@vpn5> traceroute 192.168.192.4 source 10.255.10.5
traceroute to 192.168.192.4 (192.168.192.4) from 10.255.10.5, 30 hops max, 40
byte packets
 1 vpn-08-t3003.isp-core.net (192.168.193.2) 0.678 ms 0.549 ms 0.494 ms
 2 vpn1-so-100.isp-core.net (192.168.192.100) 0.873 ms 0.847 ms 0.844 ms
   MPLS Label=100021 CoS=0 TTL=1 S=1
 3 vpn4-fe-112.isp-core.net (192.168.192.4) 0.825 ms 0.743 ms 0.764 ms
```

Ping the Directly Connected PE and CE Routers from Each Other

From the loopback interfaces on the CE routers, you can ping the VPN interface on the directly connected PE router. See Figure 15 for the topology referenced in these examples.

From the loopback interface on Router CE1 (VPN4), ping the VPN interface, fe-1/1/0.0, on Router PE1:

```
user@vpn4> ping 192.168.192.1 local 10.255.10.4 count 3
PING 192.168.192.1 (192.168.192.1): 56 data bytes
64 bytes from 192.168.192.1: icmp_seq=0 ttl=255 time=0.885 ms
64 bytes from 192.168.192.1: icmp_seq=1 ttl=255 time=0.757 ms
64 bytes from 192.168.192.1: icmp_seq=2 ttl=255 time=0.734 ms

--- 192.168.192.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.734/0.792/0.885/0.066 ms
```

To determine the path from the loopback interface on Router CE1 to the VPN interfaces on Router PE1, use the following traceroute command:

```
user@vpn4> traceroute 192.168.192.1 source 10.255.10.4
traceroute to 192.168.192.1 (192.168.192.1) from 10.255.10.4, 30 hops max, 40
byte packets
 1 vpn1-fe-110.isp-core.net (192.168.192.1) 0.828 ms 0.657 ms 1.972 ms
```

From the loopback interface on Router CE2 (VPN5), ping the VPN interface, t3-0/0/3.0, on Router PE2:

```
user@vpn5> ping 192.168.193.2 local 10.255.10.5 count 3
PING 192.168.193.2 (192.168.193.2): 56 data bytes
64 bytes from 192.168.193.2: icmp_seq=0 ttl=255 time=0.998 ms
64 bytes from 192.168.193.2: icmp_seq=1 ttl=255 time=0.834 ms
64 bytes from 192.168.193.2: icmp_seq=2 ttl=255 time=0.819 ms

--- 192.168.193.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.819/0.884/0.998/0.081 ms
```

To determine the path from the loopback interface on Router CE2 to the VPN interfaces on Router PE2, use the following traceroute command:

```
serpil@vpn5> traceroute 192.168.193.2 source 10.255.10.5
traceroute to 192.168.193.2 (192.168.193.2) from 10.255.10.5, 30 hops max, 40
byte packets
 1  vpn-08-t3003.isp-core.net (192.168.193.2)  0.852 ms  0.670 ms  0.656 ms
```

From the VPN interface on the PE router, you can ping the VPN or loopback interface on the directly connected CE router.

From the VPN interface on Router PE1 (VPN1), ping the VPN interface on Router CE1, fe-1/1/0.0:

```
user@vpn1> ping 192.168.192.4 vpn-interface fe-1/1/0.0 local 192.168.192.1 count
3
PING 192.168.192.4 (192.168.192.4): 56 data bytes
64 bytes from 192.168.192.4: icmp_seq=0 ttl=255 time=0.866 ms
64 bytes from 192.168.192.4: icmp_seq=1 ttl=255 time=0.728 ms
64 bytes from 192.168.192.4: icmp_seq=2 ttl=255 time=0.753 ms

--- 192.168.192.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.728/0.782/0.866/0.060 ms
```

From the VPN interface on Router PE1 (VPN1), ping the loopback interface on Router CE1, 10.255.10.4:

```
user@vpn1> ping 10.255.10.4 vpn-interface fe-1/1/0.0 local 192.168.192.1 count 3
PING 10.255.10.4 (10.255.10.4): 56 data bytes
64 bytes from 10.255.10.4: icmp_seq=0 ttl=255 time=0.838 ms
64 bytes from 10.255.10.4: icmp_seq=1 ttl=255 time=0.760 ms
64 bytes from 10.255.10.4: icmp_seq=2 ttl=255 time=0.771 ms

--- 10.255.10.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.760/0.790/0.838/0.034 ms
```

To determine the path from the VPN interface on Router PE1 to the VPN and loopback interfaces on Router CE1, respectively, use the following traceroute commands:

```
user@vpn1> traceroute 10.255.10.4 vpn-interface fe-1/1/0.0 source 192.168.192.1
traceroute to 10.255.10.4 (10.255.10.4) from 192.168.192.1, 30 hops max, 40 byte
packets
 1  vpn4.isp-core.net (10.255.10.4)  0.842 ms  0.659 ms  0.621 ms

user@vpn1> traceroute 192.168.192.4 vpn-interface fe-1/1/0.0 source
192.168.192.1
traceroute to 192.168.192.4 (192.168.192.4) from 192.168.192.1, 30 hops max, 40
byte packets
 1  vpn4-fe-112.isp-core.net (192.168.192.4)  0.810 ms  0.662 ms  0.640 ms
```

From the VPN interface on Router PE2 (VPN2), ping the VPN interface on Router CE2, t3-0/0/3.0:

```
user@vpn2> ping 192.168.193.5 vpn-interface t3-0/0/3.0 local 192.168.193.2
count 3
PING 192.168.193.5 (192.168.193.5): 56 data bytes
64 bytes from 192.168.193.5: icmp_seq=0 ttl=255 time=0.852 ms
64 bytes from 192.168.193.5: icmp_seq=1 ttl=255 time=0.909 ms
64 bytes from 192.168.193.5: icmp_seq=2 ttl=255 time=0.793 ms

--- 192.168.193.5 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.793/0.851/0.909/0.047 ms
```

From the VPN interface on Router PE2 (VPN2), ping the loopback interface on Router CE2, 10.255.10.5:

```
user@vpn2> ping 10.255.10.5 vpn-interface t3-0/0/3.0 local 192.168.193.2 count 3
PING 10.255.10.5 (10.255.10.5): 56 data bytes
64 bytes from 10.255.10.5: icmp_seq=0 ttl=255 time=0.914 ms
64 bytes from 10.255.10.5: icmp_seq=1 ttl=255 time=0.888 ms
64 bytes from 10.255.10.5: icmp_seq=2 ttl=255 time=1.066 ms

--- 10.255.10.5 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.888/0.956/1.066/0.079 ms
```

To determine the path from the VPN interface on Router PE2 to the VPN and loopback interfaces on Router CE2, respectively, use the following traceroute commands:

```
user@vpn2> traceroute 10.255.10.5 vpn-interface t3-0/0/3.0 source 192.168.193.2
traceroute to 10.255.10.5 (10.255.10.5) from 192.168.193.2, 30 hops max, 40 byte packets
 1  vpn5.isp-core.net (10.255.10.5)  1.009 ms  0.677 ms  0.633 ms

user@vpn2> traceroute 192.168.193.5 vpn-interface t3-0/0/3.0 source
192.168.193.2
traceroute to 192.168.193.5 (192.168.193.5) from 192.168.193.2, 30 hops max, 40
byte packets
 1  vpn5-t3-003.isp-core.net (192.168.193.5)  0.974 ms  0.665 ms  0.619 ms
```

Ping a Remote CE Router from a PE Router

There is a limitation on how you ping a remote CE router from a PE router. If there is a problem with the connection to the local CE router and you are attempting to ping a remote CE router using the default loopback address, the ping can fail. This limitation and a way to work around it are described in the following sections:

- Limitation on Pinging a Remote CE Router from a PE Router on page 124
- Configure a Logical Unit on the Loopback Interface on page 124

Limitation on Pinging a Remote CE Router from a PE Router

If you attempt to ping a remote CE router from a PE router, ICMP echo requests are sent from the PE router, with the PE router's VPN interface as the source. Other PE routers have a route back to that address with a VPN label. When the echo replies return, they include a label. The PE router pops the VPN label and sends the packet from the VPN interface to the local CE router. The local CE router sends it back to the PE router, its actual destination.

When a Juniper Networks router receives a labeled packet, the label is popped (depending on the label operation specified), and the packet is forwarded to an interface, even if the packet is destined for that particular PE router. Labeled packets are not analyzed further for the IP information under the label.

If there is a problem with the connection to the local CE router, packets are sent out but do not return to the PE router, and the ping fails. If the connection between your PE router and local CE router is down, sending a ping to the remote CE router fails even though the connection to the remote CE router might be functional.

Configure a Logical Unit on the Loopback Interface

The following procedure is effective for Layer 3 VPNs only. To ping a remote CE router from a local PE router in a Layer 3 VPN, you can configure a logical unit for the loopback interface and configure this loopback interface to the Layer 3 VPN routing instance. You can associate one logical loopback interface with each VRF routing instance, enabling you to ping a specific routing instance on a router.

To configure an additional logical unit on the loopback interface of the PE router, configure the unit statement at the [edit interfaces lo0] hierarchy level:

```
[edit interfaces]
lo0 {
  unit number {
    family inet {
      address address;
    }
  }
}
```

You then configure the logical unit on the loopback interface for the VRF routing instance on the PE router. To do this, include the interface statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit routing-instances routing-instance-name]
interface interface-name;
```

The *interface-name* is the logical unit on the loopback interface (for example, lo0.1).

From the VPN interface on PE router, you can now ping the logical unit on the loopback interface on the remote CE router:

```
ping vpn-interface vpn-interface host
```

Use *vpn-interface* to specify the new logical unit on the loopback interface (for example, lo0.1). For more information on how to use the ping vpn-interface command, see the *JUNOS Internet Software Operational Mode Command Reference*.

Disable Normal TTL Decrementing for Layer 3 VPNs

For information on how to disable normal time-to-live (TTL) decrementing for Layer 3 VPNs, see “Disable Normal TTL Decrementing for VPNs” on page 26.

Indirect Next-hop Address Space and Route Reflectors

If you attempt to allocate indirect next-hop indexes for all the active routes in the routing table, the indirect next-hop address space can be exhausted. This occurs most frequently in topologies in which a Juniper Networks router acts as a VPN route reflector and a BGP peer with other vendors' routers. The other routers advertise unique label stacks per prefix, instead of sharing common label stacks across many prefixes.

You can configure a forwarding table export policy to prevent routes from being installed in the forwarding table, even when those routes are active in the routing table. With this change, the routes that are omitted from the forwarding table do not have additional indirect next-hop indexes allocated to them.

The following configuration avoids indirect next-hop address space exhaustion, but does not allow the router to forward traffic for the BGP learned prefixes:

```
routing-options {
  forwarding-table {
    export kern;
  }
}
policy-options {
  policy-statement kern {
    from protocol bgp;
    then reject;
  }
}
```

.....

Chapter 9

Layer 3 VPN Configuration Examples

This chapter provides the following examples of Layer 3 virtual private network (VPN) configurations:

- Configure a Simple Full-Mesh VPN Topology on page 128
- Configure a Full-Mesh VPN Topology with Route Reflectors on page 142
- Configure a Hub-and-Spoke VPN Topology on page 142
- Configure an LDP-over-RSVP VPN Topology on page 158
- Configure an Application-Based Layer 3 VPN Topology on page 172
- Configure an OSPF Domain ID for a Layer 3 VPN on page 177
- Configure Overlapping VPNs Using Routing Table Groups on page 183
- Configuring Overlapping VPNs Using Automatic Route Export on page 195
- Configure a GRE Tunnel Interface between PE Routers on page 198
- Configure a GRE Tunnel Interface between a PE and CE Router on page 205
- Configure an ES Tunnel Interface between a PE and CE Router on page 208



Note

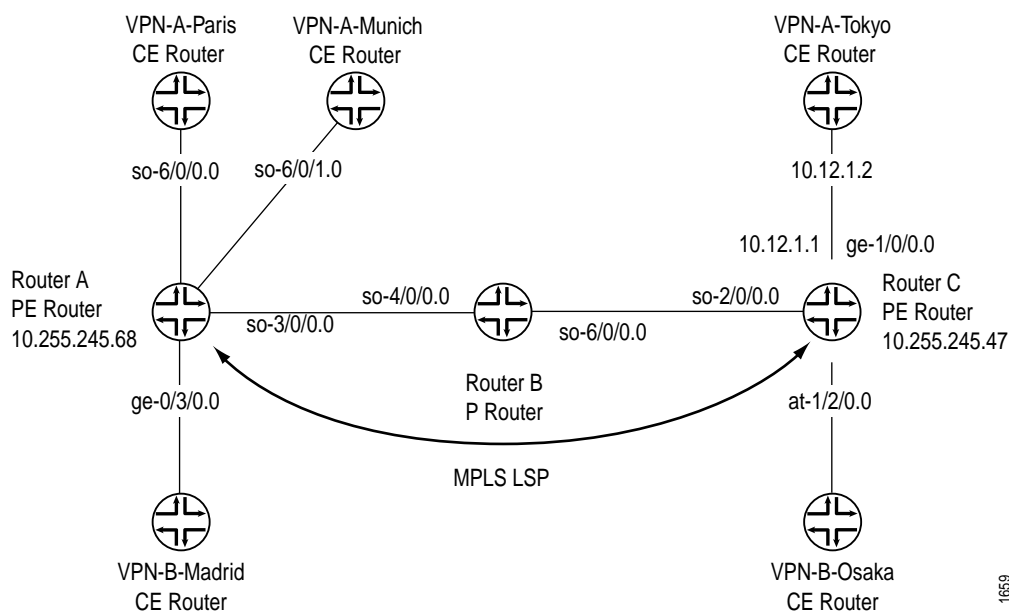
The examples in this chapter show only the portions of the configuration that establish VPN functionality. You must also configure other router functionality, including all router interfaces, for a router configuration to work properly.

Configure a Simple Full-Mesh VPN Topology

This example shows how to set up a simple full-mesh service provider VPN configuration, which consists of the following components (see Figure 16):

- Two separate VPNs (VPN-A and VPN-B)
- Two provider edge (PE) routers, both of which service VPN-A and VPN-B
- Resource Reservation Protocol (RSVP) as the signaling protocol
- One RSVP label-switched path (LSP) that tunnels between the two PE routers through one provider (P) router

Figure 16: Example of a Simple VPN Topology



In this configuration, route distribution in VPN A from the router VPN-A-Paris to the router VPN-A-Tokyo occurs as follows:

1. The customer edge (CE) router VPN-A-Paris announces routes to the PE router Router A.
2. Router A installs the received announced routes into its VPN routing and forwarding (VRF) table, VPN-A.inet.0.
3. Router A creates a Multiprotocol Label Switching (MPLS) label for the interface between it and the router VPN-A-Paris.
4. Router A checks its VRF export policy.
5. Router A converts the Internet Protocol Version 4 (IPv4) routes from VPN-A-Paris into VPN IPv4 format using its route distinguisher and announces these routes to PE Router C over the internal Border Gateway Protocol (IBGP) between the two PE routers.

6. Router C checks its VRF import policy and installs all routes that match the policy into its bgp.13vpn.0 routing table. (Any routes that do not match are discarded.)
7. Router C checks its VRF import policy and installs all routes that match into its VPN-A.inet.0 routing table. The routes are installed in IPv4 format.
8. Router C announces its routes to the CE router VPN-A-Tokyo, which installs them into its master routing table. (For routers running JUNOS software, the master routing table is inet.0.)
9. Router C uses the LSP between it and Router A to route all packets from router VPN-A-Tokyo that are destined for the router VPN-A-Paris.

The following sections explain how to configure the VPN functionality on the PE and provider routers. The CE routers are not aware of the VPN, so you configure them normally.

- Enable an IGP on the PE and Provider Routers on page 129
- Enable RSVP and MPLS on the Provider Router on page 130
- Configure the MPLS LSP Tunnel between the PE Routers on page 130
- Configure IBGP on the PE Routers on page 131
- Configure Routing Instances for VPNs on the PE Routers on page 132
- Configure VPN Policy on the PE Routers on page 134

The final section in this example, “Simple VPN Configuration Summarized by Router” on page 137, consolidates the statements needed to configure VPN functionality on each of the service provider routers shown in Figure 16.



Note

In this example, a private autonomous system (AS) number is used for the route distinguisher and the route target. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

Enable an IGP on the PE and Provider Routers

To allow the PE and provider routers to exchange routing information among themselves, you must configure an interior gateway protocol (IGP) on all these routers or you must configure static routes. You configure the IGP on the master instance of the routing protocol process (rpd) (that is, at the [edit protocols] hierarchy level), not within the VPN routing instance (that is, not at the [edit routing-instances] hierarchy level).

You configure the IGP in the standard way. This configuration example does not include this portion of the configuration.

Enable RSVP and MPLS on the Provider Router

On the provider router, Router B, you must configure RSVP and MPLS because this router exists on the MPLS LSP path between the two PE routers, Router A and Router C:

```
[edit]
protocols {
  rsvp {
    interface so-4/0/0.0;
    interface so-6/0/0.0;
  }
  mpls {
    interface so-4/0/0.0;
    interface so-6/0/0.0;
  }
}
```

Configure the MPLS LSP Tunnel between the PE Routers

In this configuration example, RSVP is used for VPN signaling. Therefore, in addition to configuring RSVP, you must enable traffic engineering support in an IGP and you must create an MPLS LSP to tunnel the VPN traffic.

On PE Router A, enable RSVP and configure one end of the MPLS LSP tunnel. In this example, traffic engineering support is enabled for Open Shortest Path First (OSPF). When configuring the MPLS LSP, include interface statements for all interfaces participating in MPLS, including the interfaces to the PE and CE routers. The statements for the interfaces between the PE and CE routers are needed so that the PE router can create an MPLS label for the private interface. In this example, the first interface statement configures MPLS on the interface connected to the LSP, and the remaining three configure MPLS on the interfaces that connect the PE router to the CE routers.

```
[edit]
protocols {
  rsvp {
    interface so-3/0/0.0;
  }
  mpls {
    label-switched-path RouterA-to-RouterC {
      to 10.255.245.47;
    }
    interface so-3/0/0.0;
    interface so-6/0/0.0;
    interface so-6/0/1.0;
    interface ge-0/3/0.0;
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface so-3/0/0.0;
    }
  }
}
```

On PE Router C, enable RSVP and configure the other end of the MPLS LSP tunnel. Again, traffic engineering support is enabled for OSPF, and you configure MPLS on the interfaces to the LSP and the CE routers.

```
[edit]
protocols {
  rsvp {
    interface so-2/0/0.0;
  }
  mpls {
    label-switched-path RouterC-to-RouterA {
      to 10.255.245.68;
    }
    interface so-2/0/0.0;
    interface ge-1/0/0.0;
    interface at-1/2/0.0;
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface so-2/0/0.0;
    }
  }
}
```

Configure IBGP on the PE Routers

On the PE routers, configure an IBGP session with the following properties:

- **VPN family**—To indicate that the IBGP session is for the VPN, include the family inet-vpn statement.
- **Loopback address**—Include the local-address statement, specifying the local PE router's loopback address. The IBGP session for VPNs runs through the loopback address. Note that you must also configure the lo0 interface at the [edit interfaces] hierarchy level. The example does not include this part of the router's configuration.
- **Neighbor address**—Include the neighbor statement, specifying the IP address of the neighboring PE router, which is its loopback (lo0) address.

On PE Router A, configure IBGP as follows:

```
[edit]
protocols {
  bgp {
    group PE-RouterA-to-PE-RouterC {
      type internal;
      local-address 10.255.245.68;
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.245.47;
    }
  }
}
```

On PE Router C, configure IBGP as follows:

```
[edit]
protocols {
  bgp {
    group PE-RouterC-to-PE-RouterA {
      type internal;
      local-address 10.255.245.47;
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.245.68;
    }
  }
}
```

Configure Routing Instances for VPNs on the PE Routers

Both PE routers service VPN-A and VPN-B, so you must configure two routing instances on each router, one for each VPN. For each VPN, you must define the following in the routing instance:

- Route distinguisher, which must be unique for each routing instance on the PE router. It is used to distinguish the addresses in one VPN from those in another VPN.
- Instance type of vrf, which creates the VRF table on the PE router.
- Interfaces connected to the CE routers.
- VRF import and export policies, which must be the same on each PE router that services the same VPN. Unless an import policy contains only a then reject statement, it must include reference to a community. Otherwise, when you try to commit the configuration, the commit fails.



Note

In this example, a private AS number is used for the route distinguisher. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

- Routing between the PE and CE routers, which is required for the PE router to distribute VPN-related routes to and from connected CE routers. You can configure a routing protocol—Border Gateway Protocol (BGP), OSPF, or Routing Information Protocol (RIP)—or you can configure static routing.

On PE Router A, configure the following routing instance for VPN-A. In this example, Router A uses static routes to distribute routes to and from the two CE routers to which it is connected.

```
[edit]
routing-instance {
  VPN-A-Paris-Munich {
    instance-type vrf;
    interface so-6/0/0.0;
    interface so-6/0/1.0;
    route-distinguisher 65535:0;
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
    routing-options {
      static {
        route 172.16.0.0/16 next-hop so-0/0/0.0;
        route 172.17.0.0/16 next-hop so-6/0/1.0;
      }
    }
  }
}
```

On PE Router C, configure the following routing instance for VPN-A. In this example, Router C uses BGP to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
  VPN-A-Tokyo {
    instance-type vrf;
    interface ge-1/0/0.0;
    route-distinguisher 65535:1;
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
    protocols {
      bgp {
        group VPN-A-Site2 {
          peer-as 1;
          neighbor 10.12.1.2;
        }
      }
    }
  }
}
```

On PE Router A, configure the following routing instance for VPN-B. In this example, Router A uses OSPF to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
  VPN-B-Madrid {
    instance-type vrf;
    interface ge-0/3/0.0;
    route-distinguisher 65535:2;
    vrf-import VPN-B-import;
    vrf-export VPN-B-export;
    protocols {
      ospf {
        export bgp-to-ospf;
        area 0.0.0.0 {
          interface ge-0/3/0;
        }
      }
    }
  }
}
```

On PE Router C, configure the following routing instance for VPN-B. In this example, Router C uses RIP to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
  VPN-B-Osaka {
    instance-type vrf;
    interface at-1/2/0.0;
    route-distinguisher 65535:3;
    vrf-import VPN-B-import;
    vrf-export VPN-B-export;
    protocols {
      rip {
        group PE-C-to-VPN-B {
          export bgp-to-rip;
          neighbor at-1/2/0;
        }
      }
    }
  }
}
```

Configure VPN Policy on the PE Routers

You must configure VPN import and export policies on each of the PE routers so that they install the appropriate routes in their VRF tables, which they use to forward packets within a VPN. For VPN-A, the VRF table is VPN-A.inet.0, and for VPN-B it is VPN-B.inet.0.

In the VPN policy, you also configure VPN target communities.



Note

In this example, a private AS number is used for the route target. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

On PE Router A, configure the following VPN import and export policies.



Note

The policy qualifiers shown in this example are only those needed for the VPN to function. You can configure additional qualifiers, as needed, to any policies that you configure.

```
[edit]
policy-options {
  policy-statement VPN-A-import {
    term a {
      from {
        protocol bgp;
        community VPN-A;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-A-export {
    term a {
      from protocol static;
      then {
        community add VPN-A;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-B-import {
    term a {
      from {
        protocol bgp;
        community VPN-B;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-B-export {
    term a {
      from protocol ospf;
      then {
        community add VPN-B;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community VPN-A members target:65535:0;
  community VPN-B members target:65535:2;
}
```

On PE Router C, configure the following VPN import and export policies:

```
[edit]
policy-options {
  policy-statement VPN-A-import {
    term a {
      from {
        protocol bgp;
        community VPN-A;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-A-export {
    term a {
      from protocol bgp;
      then {
        community add VPN-A;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-B-import {
    term a {
      from {
        protocol bgp;
        community VPN-B;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-B-export {
    term a {
      from protocol rip;
      then {
        community add VPN-B;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community VPN-A members target:65535:1;
  community VPN-B members target:65535:3;
}
```

To apply the VPN policies on the routers, include the `vrf-export` and `vrf-import` statements when you configure the routing instance. For both VPNs, the VRF import and export policies handle the route distribution across the IBGP session running between the PE routers.

To apply the VPN policies on PE Router A, include the following statements:

```
[edit]
routing-instance {
  VPN-A-Paris-Munich {
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
  }
  VPN-B-Madrid {
    vrf-import VPN-B-import;
    vrf-export VPN-B-export;
  }
}
```

To apply the VPN policies on PE Router C, include the following statements:

```
[edit]
routing-instance {
  VPN-A-Tokyo {
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
  }
  VPN-B-Osaka {
    vrf-import VPN-B-import;
    vrf-export VPN-B-export;
  }
}
```

Simple VPN Configuration Summarized by Router

Router A (PE Router)

Routing Instance for VPN-A	<pre>routing-instance { VPN-A-Paris-Munich { instance-type vrf; interface so-6/0/0.0; interface so-6/0/1.0; route-distinguisher 65535:0; vrf-import VPN-A-import; vrf-export VPN-A-export; } }</pre>
Instance Routing Protocol	<pre>routing-options { static { route 172.16.0.0/16 next-hop so-6/0/0.0; route 172.17.0.0/16 next-hop so-6/0/1.0; } }</pre>
Routing Instance for VPN-B	<pre>routing-instance { VPN-B-Madrid { instance-type vrf; interface ge-0/3/0.0; route-distinguisher 65535:2; vrf-import VPN-B-import; vrf-export VPN-B-export; } }</pre>

```

Instance Routing Protocol      protocols {
                                ospf {
                                    area 0.0.0.0 {
                                        interface ge-0/3/0;
                                    }
                                }
                            }

Master Protocol Instance protocols {

                                Enable RSVP      rsvp {
                                    interface so-3/0/0.0;
                                }

                                Configure an MPLS LSP mpls {
                                    label-switched-path RouterA-to-RouterC {
                                        to 10.255.245.47;
                                    }
                                    interface so-3/0/0.0;
                                    interface so-6/0/0.0;
                                    interface so-6/0/1.0;
                                    interface ge-0/3/0.0;
                                }

                                Configure IBGP      bgp {
                                    group PE-RouterA-to-PE-RouterC {
                                        type internal;
                                        local-address 10.255.245.68;
                                        family inet-vpn {
                                            unicast;
                                        }
                                        neighbor 10.255.245.47;
                                    }
                                }

                                Configure OSPF for Traffic Engineering Support ospf {
                                    traffic-engineering;
                                    area 0.0.0.0 {
                                        interface so-3/0/0.0;
                                    }
                                }

                                Configure VPN Policy policy-options {
                                    policy-statement VPN-A-import {
                                        term a {
                                            from {
                                                protocol bgp;
                                                community VPN-A;
                                            }
                                            then accept;
                                        }
                                        term b {
                                            then reject;
                                        }
                                    }
                                }

```

```

policy-statement VPN-A-export {
  term a {
    from protocol static;
    then {
      community add VPN-A;
      accept;
    }
  }
  term b {
    then reject;
  }
}
policy-statement VPN-B-import {
  term a {
    from {
      protocol bgp;
      community VPN-B;
    }
    then accept;
  }
  term b {
    then reject;
  }
}
policy-statement VPN-B-export {
  term a {
    from protocol ospf;
    then {
      community add VPN-B;
      accept;
    }
  }
  term b {
    then reject;
  }
}
community VPN-A members target:65535:0;
community VPN-B members target:65535:2;
}

```

Router B (Provider Router)

```

Master Protocol Instance protocols {
  Enable RSVP rsvp {
    interface so-4/0/0.0;
    interface so-6/0/0.0;
  }
  Enable MPLS mpls {
    interface so-4/0/0.0;
    interface so-6/0/0.0;
  }
}

```

Router C (PE Router)

```

Routing Instance for VPN-A  routing-instance {
                               VPN-A-Tokyo {
                               instance-type vrf;
                               interface ge-1/0/0.0;
                               route-distinguisher 65535:1;
                               vrf-import VPN-A-import;
                               vrf-export VPN-A-export;

Instance Routing Protocol   protocols {
                               bgp {
                               group VPN-A-Site2 {
                               peer-as 1;
                               neighbor 10.12.1.2;
                               }
                               }
                               }

Routing Instance for VPN-B  VPN-B-Osaka {
                               instance-type vrf;
                               interface at-1/2/0.0;
                               route-distinguisher 65535:3;
                               vrf-import VPN-B-import;
                               vrf-export VPN-B-export;

Instance Routing Protocol   protocols {
                               rip {
                               group PE-C-to-VPN-B {
                               neighbor at-1/2/0;
                               }
                               }
                               }

Master Protocol Instance    protocols {

Enable RSVP                 rsvp {
                               interface so-2/0/0.0;
                               }

Configure an MPLS LSP      mpls {
                               label-switched-path RouterC-to-RouterA {
                               to 10.255.245.68;
                               }
                               interface so-2/0/0.0;
                               interface ge-1/0/0.0;
                               interface at-1/2/0.0;
                               }

Configure IBGP             bgp {
                               group PE-RouterC-to-PE-RouterA {
                               type internal;
                               local-address 10.255.245.47;
                               family inet-vpn {
                               unicast;
                               }
                               neighbor 10.255.245.68;
                               }
                               }

```

Configure OSPF for Traffic Engineering Support

```
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface so-2/0/0.0;
  }
}
```

Configure VPN Policy

```
policy-options {
  policy-statement VPN-A-import {
    term a {
      from {
        protocol bgp;
        community VPN-A;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-A-export {
    term a {
      from protocol bgp;
      then {
        community add VPN-A;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-B-import {
    term a {
      from {
        protocol bgp;
        community VPN-B;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-B-export {
    term a {
      from protocol rip;
      then {
        community add VPN-B;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community VPN-A members target:65535:1;
  community VPN-B members target:65535:3;
}
```

Configure a Full-Mesh VPN Topology with Route Reflectors

This example is a variation of the full-mesh VPN topology example (described in “Configure a Simple Full-Mesh VPN Topology” on page 128) in which one of the PE routers is a BGP route reflector. In this variation, Router C in Figure 16 on page 128 is a route reflector. The only change to its configuration is that you need to include the cluster statement when configuring the BGP group:

```
[edit protocols]
bgp {
  group PE-RouterC-to-PE-RouterA {
    type internal;
    local-address 10.255.245.47;
    family inet-vpn {
      unicast;
    }
    neighbor 10.255.245.68;
    cluster 4.3.2.1;
  }
}
```

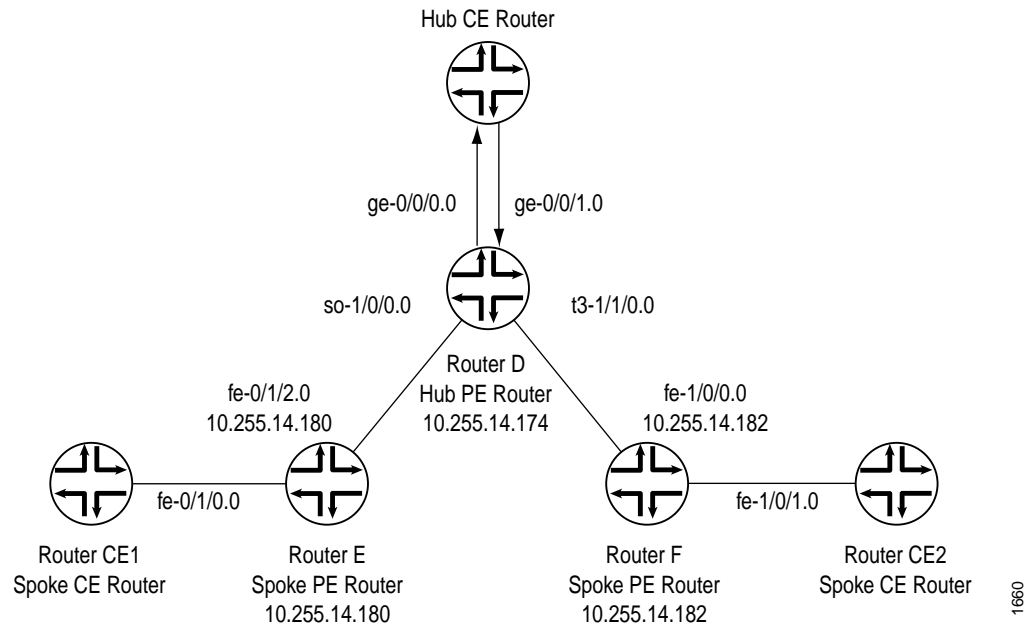
For the complete configuration example of Router C, see “Router C (PE Router)” on page 140.

Configure a Hub-and-Spoke VPN Topology

This example shows how to set up a hub-and-spoke VPN configuration, which consists of the following components (see Figure 17):

- One hub PE router (Router D).
- One hub CE router connected to the hub PE router. For a hub-and-spoke VPN topology to function properly, there must be two interfaces connecting the hub PE router to the hub CE router, and each interface must have its own VRF table on the PE router:
 - One interface (here, interface ge-0/0/0.0) is used to announce spoke routes to the hub CE router. The VRF table associated with this interface contains the routes being announced by the spoke PE routers to the hub CE router.
 - The second interface (here, interface ge-0/0/1.0) is used to receive route announcements from the hub CE that are destined for the hub-and-spoke routers. The VRF table associated with this interface contains the routes announced by the hub CE router to the spoke PE routers.
- Two spoke PE routers (Router E and Router F).
- Two spoke CE routers (CE1 and CE2), one connected to each spoke PE router.
- Label Distribution Protocol (LDP) as the signaling protocol.

Figure 17: Example of a Hub-and-Spoke VPN Topology

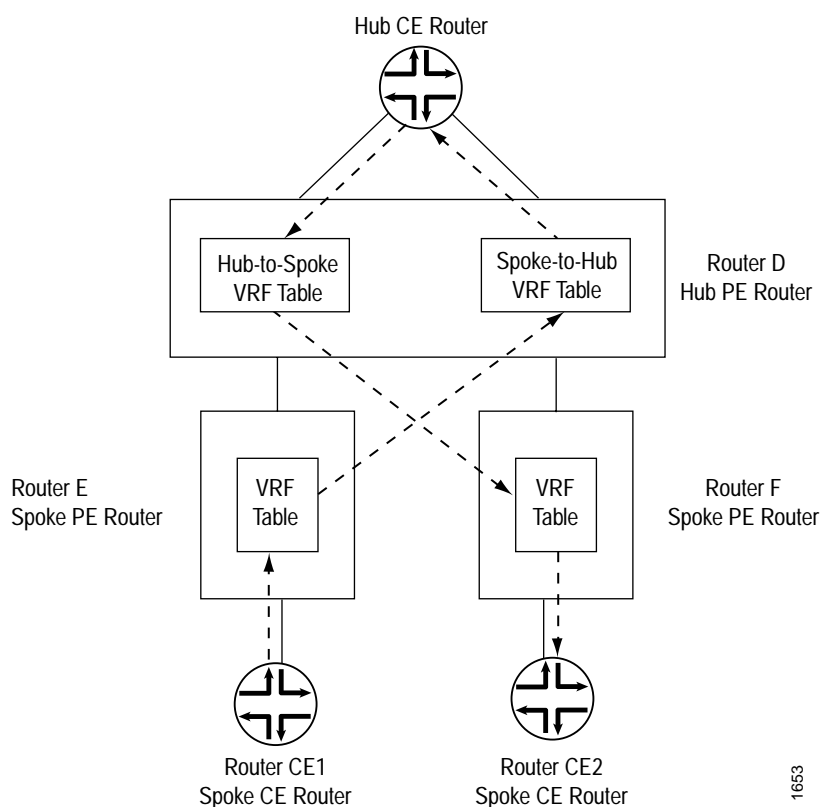


In this configuration, route distribution from spoke CE Router CE1 occurs as follows:

1. Spoke Router CE1 announces its routes to spoke PE Router E.
2. Router E installs the routes from CE1 into its VRF table.
3. After checking its VRF export policy, Router E adds the spoke target community to the routes from Router CE1 that passed the policy and announces them to the hub PE router, Router D.
4. Router D checks the VRF import policy associated with interface ge-0/0/0.0 and places all routes from spoke PE routers that match the policy into its bgp.l3vpn routing table. (Any routes that do not match are discarded.)
5. Router D checks its VRF import policy associated with interface ge-0/0/0.0 and installs all routes that match into its spoke VRF table. The routes are installed with the spoke target community.
6. Router D announces routes to the hub CE over interface ge-0/0/0.
7. The hub CE router announces the routes back to the hub PE Router D over the second interface to the hub router, interface ge-0/0/1.
8. The hub PE router installs the routes learned from the hub CE router into its hub VRF table, which is associated with interface ge-0/0/1.
9. The hub PE router checks the VRF export policy associated with interface ge-0/0/1.0 and announces all routes that match to all spokes after adding the hub target community.

Figure 18 illustrates how routes are distributed from this spoke router to the other spoke CE router, Router CE2. The same path is followed if you issue a traceroute command from Router CE1 to Router CE2.

Figure 18: Route Distribution between Two Spoke Routers



The following sections explain how to configure the VPN functionality for a hub-and-spoke topology on the hub-and-spoke PE routers. The CE routers do not know about the VPN, so you configure them normally.

- Enable an IGP on the Hub-and-Spoke PE Routers on page 145
- Configure LDP on the Hub-and-Spoke PE Routers on page 145
- Configure IBGP on the PE Routers on page 146
- Configure Routing Instances for VPNs on the Hub-and-Spoke PE Routers on page 147
- Configure VPN Policy on the PE Routers on page 150

The final section in this example, “Hub-and-Spoke VPN Configuration Summarized by Router” on page 153, consolidates the statements needed to configure VPN functionality for each of the service provider routers shown in Figure 17.

Enable an IGP on the Hub-and-Spoke PE Routers

To allow the hub-and-spoke PE routers to exchange routing information, you must configure an IGP on all these routers or you must configure static routes. You configure the IGP on the master instance of the routing protocol process (rpd) (that is, at the [edit protocols] hierarchy level), not within the routing instance (that is, not at the [edit routing-instances] hierarchy level).

You configure the IGP in the standard way. This configuration example does not include this portion of the configuration.

In the route distribution in a hub-and-spoke topology, if the protocol used between the CE and PE routers at the hub site is BGP, the hub CE router announces all routes received from the hub PE router and the spoke routers back to the hub PE router and all the spoke routers. This means that the hub-and-spoke PE routers receive routes that contain their AS number. Normally, when a route contains this information, it indicates that a routing loop has occurred and the router rejects the routes. However, for the VPN configuration to work, the hub PE router and the spoke routers must accept these routes. To enable this, include the loops option when configuring the AS at the [edit routing-options] hierarchy level on the hub PE router and all the spoke routers. For this example configuration, you specify a value of 1. You can specify a number from 0 through 10.

```
[edit routing-options]
autonomous-system as-number loops 1;
```

Configure LDP on the Hub-and-Spoke PE Routers

You must configure LDP on the interfaces between the hub-and-spoke PE routers that participate in the VPN.

On hub PE Router D, configure LDP as follows:

```
[edit protocols]
ldp {
  interface so-1/0/0.0;
  interface t3-1/1/0.0;
}
```

On spoke PE Router E, configure LDP as follows:

```
[edit protocols]
ldp {
  interface fe-0/1/2.0;
}
```

On spoke PE router F, configure LDP as follows:

```
[edit protocols]
ldp {
  interface fe-1/0/0.0;
}
```

Configure IBGP on the PE Routers

On the hub-and-spoke PE routers, configure an IBGP session with the following properties:

- **VPN family**—To indicate that the IBGP session is for the VPN, include the family inet-vpn statement.
- **Loopback address**—Include the local-address statement, specifying the local PE router's loopback address. The IBGP session for VPNs runs through the loopback address. Note that you must also configure the lo0 interface at the [edit interfaces] hierarchy level. The example does not include this part of the router's configuration.
- **Neighbor address**—Include the neighbor statement. On the hub router, specify the IP address of each spoke PE router, and on the spoke router, specify the address of the hub PE router.

For the hub router, you configure an IBGP session with each spoke, and for each spoke router, you configure an IBGP session with the hub. There are no IBGP sessions between the two spoke routers.

On hub Router D, configure IBGP as follows. The first neighbor statement configures an IBGP session to spoke Router E, and the second configures a session to spoke Router F.

```
[edit protocols]
bgp {
  group Hub-to-Spokes {
    type internal;
    local-address 10.255.14.174;
    family inet-vpn {
      unicast;
    }
    neighbor 10.255.14.180;
    neighbor 10.255.14.182;
  }
}
```

On spoke Router E, configure an IBGP session to the hub router as follows:

```
[edit protocols]
bgp {
  group Spoke-E-to-Hub {
    type internal;
    local-address 10.255.14.180;
    neighbor 10.255.14.174 {
      family inet-vpn {
        unicast;
      }
    }
  }
}
```

On spoke Router F, configure an IBGP session to the hub router as follows:

```
[edit protocols]
bgp {
  group Spoke-F-to-Hub {
    type internal;
    local-address 10.255.14.182;
    neighbor 10.255.14.174 {
      family inet-vpn {
        unicast;
      }
    }
  }
}
```

Configure Routing Instances for VPNs on the Hub-and-Spoke PE Routers

For the hub PE router to be able to distinguish between packets going to and coming from the spoke PE routers, you must configure it with two routing instances:

- One routing instance (in this example, Spokes-to-Hub-CE) is associated with the interface that carries packets from the hub PE router to the hub CE router (in this example, interface ge-0/0/0.0). Its VRF table contains the routes being announced by the spoke PE routers and the hub PE router to the hub CE router.
- The second routing instance (in this example, Hub-CE-to-Spokes) is associated with the interface that carries packets from the hub CE router to the hub PE router (in this example, interface ge-0/0/1.0). Its VRF table contains the routes being announced from the hub CE router to the hub-and-spoke PE routers.

On each spoke router, you must configure one routing instance.

You must define the following in the routing instance:

- Route distinguisher, which is used to distinguish the addresses in one VPN from those in another VPN.
- Instance type of vrf, which creates the VRF table on the PE router.
- Interfaces that are part of the VPN and that connect the PE routers to their CE routers.
- VRF import and export policies. Both import policies must include reference to a community. Otherwise, when you try to commit the configuration, the commit fails. (The exception to this is if the import policy contains only a then reject statement.) In the VRF export policy, spoke PE routers attach the spoke target community.
- Routing between the PE and CE routers, which is required for the PE router to distribute VPN-related routes to and from connected CE routers. You can configure a routing protocol—BGP, OSPF, or RIP—or you can configure static routing.

For a hub-and-spoke topology, you must configure different policies in each routing instance on the hub CE router. For the routing instance associated with the interface that carries packets from the hub PE router to the hub CE router (in this example, Spokes-to-Hub-CE), the import policy must accept all routes received on the IBGP session between the hub-and-spoke PE routers and the export policy must reject all routes received from the hub CE router. For the routing instance associated with the interfaces that carries packets from the hub CE router to the hub PE router (in this example, Hub-CE-to-Spokes), the import policy must reject all routes received from the spoke PE routers, and the export policy must export to all the spoke routers.

On hub PE Router D, configure the following routing instances. Router D uses OSPF to distribute routes to and from the hub CE router.

```
[edit]
routing-instance {
  Spokes-to-Hub-CE {
    instance-type vrf;
    interface ge-0/0/0.0;
    route-distinguisher 10.255.1.174:65535;
    vrf-import spoke;
    vrf-export null;
    protocols {
      ospf {
        export redistribute-vpn;
        area 0.0.0.0 {
          interface ge-0/0/0;
        }
      }
    }
  }
  Hub-CE-to-Spokes {
    instance-type vrf;
    interface ge-0/0/1.0;
    route-distinguisher 10.255.1.174:65535;
    vrf-import null;
    vrf-export hub;
    protocols {
      ospf {
        export redistribute-vpn;
        area 0.0.0.0 {
          interface ge-0/0/1.0;
        }
      }
    }
  }
}
```

On spoke PE Router E, configure the following routing instances. Router E uses OSPF to distribute routes to and from the spoke CE Router CE1.

```
[edit]
routing-instance {
  Spoke-E-to-Hub {
    instance-type vrf;
    interface fe-0/1/0.0;
    route-distinguisher 10.255.14.80:65535;
    vrf-import hub;
    vrf-export spoke;
    protocols {
      ospf {
        export redistribute-vpn;
        area 0.0.0.0 {
          interface fe-0/1/0.0;
        }
      }
    }
  }
}
```

On spoke PE Router F, configure the following routing instances. Router F uses OSPF to distribute routes to and from the spoke CE Router CE2.

```
[edit]
routing-instance {
  Spoke-F-to-Hub {
    instance-type vrf;
    interface fe-1/0/1.0;
    route-distinguisher 10.255.14.182:65535;
    vrf-import hub;
    vrf-export spoke;
    protocols {
      ospf {
        export redistribute-vpn;
        area 0.0.0.0 {
          interface fe-1/0/1.0;
        }
      }
    }
  }
}
```

Configure VPN Policy on the PE Routers

You must configure VPN import and export policies on each of the hub-and-spoke PE routers so that they install the appropriate routes in the VRF tables, which they use to forward packets within each VPN.

On the spoke routers, you define policies to exchange routes with the hub router.

On the hub router, you define policies to accept routes from the spoke PE routers and distribute them to the hub CE router, and vice versa. The hub PE router has two VRF tables:

- Spoke-to-hub VRF table—Handles routes received from spoke routers and announces these routes to the hub CE router. For this VRF table, the import policy must check that the spoke target name is present and that the route was received from the IBGP session between the hub PE and the spoke PE routers. This VRF table must not export any routes, so its export policy should reject everything.
- Hub-to-spoke VRF table—Handles routes received from the hub CE router and announces them to the spoke routers. For this VRF table, the export policy must add the hub target community. This VRF table must not import any routes, so its import policy should reject everything.

In the VPN policy, you also configure the VPN target communities.

On hub PE Router D, configure the following policies to apply to the VRF tables:

- spoke—Accepts routes received from the IBGP session between it and the spoke PE routers that contain the community target spoke, and rejects all other routes.
- hub—Adds the community target hub to all routes received from OSPF (that is, from the session between it and the hub CE router). It rejects all other routes.
- null—Rejects all routes.
- redistribute-vpn—Redistributes OSPF routes to neighbors within the routing instance.

```
[edit]
policy-options {
  policy-statement spoke {
    term a {
      from {
        protocol bgp;
        community spoke;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
}
```

```

policy-statement hub {
  term a {
    from protocol ospf;
    then {
      community add hub;
      accept;
    }
  }
  term b {
    then reject;
  }
}
policy-statement null {
  then reject;
}
policy-statement redistribute-vpn {
  term a {
    from protocol bgp;
    then accept;
  }
  term b {
    then reject;
  }
}
community hub members target:65535:1;
community spoke members target:65535:2;
}

```

To apply the VRF policies on Router D, include the `vrf-export` and `vrf-import` statements when you configure the routing instances:

```

[edit]
routing-instance {
  Spokes-to-Hub-CE {
    vrf-import spoke;
    vrf-export null;
  }
  Hub-CE-to-Spokes {
    vrf-import null;
    vrf-export hub;
  }
}

```

On spoke PE Router E and Router F, configure the following policies to apply to the VRF tables:

- **hub**—Accepts routes received from the IBGP session between it and the hub PE routers that contain the community target `hub`, and rejects all other routes.
- **spoke**—Adds the community target `spoke` to all routes received from OSPF (that is, from the session between it and the hub CE router) and rejects all other routes.
- **redistribute-vpn**—Redistributes OSPF routes to neighbors within the routing instance.

On spoke PE Router E and Router F, configure the following VPN import and export policies:

```
[edit]
policy-options {
  policy-statement hub {
    term a {
      from {
        protocol bgp;
        community hub;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement spoke {
    term a {
      from protocol ospf;
      then {
        community add spoke;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  policy-statement redistribute-vpn {
    term a {
      from protocol bgp;
      then accept;
    }
    term b {
      then reject;
    }
  }
  community hub members target:65535:1;
  community spoke members target 65535:2;
}
```

To apply the VRF policies on the spoke routers, include the vrf-export and vrf-import statements when you configure the routing instances:

```
[edit]
routing-instance {
  Spoke-E-to-Hub {
    vrf-import hub;
    vrf-export spoke;
  }
}

[edit]
routing-instance {
  Spoke-F-to-Hub {
    vrf-import hub;
    vrf-export spoke;
  }
}
```


Hub-and-Spoke VPN Configuration Summarized by Router

Router D (Hub PE Router)

Routing Instance for Distributing Spoke Routes to Hub CE	<pre> routing-instance { Spokes-to-Hub-CE { instance-type vrf; interface ge-0/0/0.0; route-distinguisher 10.255.1.174:65535; vrf-import spoke; vrf-export null; } }</pre>
Instance Routing Protocol	<pre> protocols { ospf { export redistribute-vpn; area 0.0.0.0 { interface ge-0/0/0; } } }</pre>
Routing Instance for Distributing Hub CE Routes to Spokes	<pre> Hub-CE-to-Spokes { instance-type vrf; interface ge-0/0/1.0; route-distinguisher 10.255.1.174:65535; vrf-import null; vrf-export hub; }</pre>
Instance Routing Protocols	<pre> protocols { ospf { export redistribute-vpn; area 0.0.0.0 { interface ge-0/0/1.0; } } }</pre>
Routing Options (Master Instance)	<pre> routing-options { autonomous-system 1 loops 1; }</pre>
Protocols (Master Instance)	<pre> protocols {</pre>
Enable LDP	<pre> ldp { interface so-1/0/0.0; interface t3-1/1/0.0; } }</pre>

```

Configure IBGP  bgp {
                  group Hub-to-Spokes {
                    type internal;
                    local-address 10.255.14.174;
                    family inet-vpn {
                      unicast;
                    }
                    neighbor 10.255.14.180;
                    neighbor 10.255.14.182;
                  }
                }

```

```

Configure VPN Policy  policy-options {
                      policy-statement spoke {
                        term a {
                          from {
                            protocol bgp;
                            community spoke;
                          }
                          then accept;
                        }
                        term b {
                          then reject;
                        }
                      }
                      policy-statement hub {
                        term a {
                          from protocol ospf;
                          then {
                            community add hub;
                            accept;
                          }
                        }
                        term b {
                          then reject;
                        }
                      }
                      policy-statement null {
                        then reject;
                      }
                      policy-statement redistribute-vpn {
                        term a {
                          from protocol bgp;
                          then accept;
                        }
                        term b {
                          then reject;
                        }
                      }
                      community hub members target:65535:1;
                      community spoke members target:65535:2;
                    }

```

Router E (Spoke PE Router)

Routing Instance	<pre> routing-instance { Spoke-E-to-Hub { instance-type vrf; interface fe-0/1/0.0; route-distinguisher 10.255.14.80:65535; vrf-import hub; vrf-export spoke; } } </pre>
Instance Routing Protocol	<pre> protocols { ospf { export redistribute-vpn; area 0.0.0.0 { interface fe-0/1/0.0; } } } </pre>
Routing Options (Master Instance)	<pre> routing-options { autonomous-system 1 loops 1; } </pre>
Protocols (Master Instance)	<pre> protocols { </pre>
Enable LDP	<pre> ldp { interface fe-0/1/2.0; } </pre>
Configure IBGP	<pre> bgp { group Spoke-E-to-Hub { type internal; local-address 10.255.14.180; neighbor 10.255.14.174 { family inet-vpn { unicast; } } } } } </pre>
Configure VPN Policy	<pre> policy-options { policy-statement hub { term a { from { protocol bgp; community hub; } then accept; } term b { then reject; } } } </pre>

```

policy-statement spoke {
  term a {
    from protocol ospf;
    then {
      community add spoke;
      accept;
    }
  }
  term b {
    then reject;
  }
}
policy-statement redistribute-vpn {
  term a {
    from protocol bgp;
    then accept;
  }
  term b {
    then reject;
  }
}
community hub members target:65535:1;
community spoke members target:65535:2;
}

```

Router F (Spoke PE Router)

```

Routing Instance  routing-instance {
                  Spoke-F-to-Hub {
                    instance-type vrf;
                    interface fe-1/0/1.0;
                    route-distinguisher 10.255.14.182:65535;
                    vrf-import hub;
                    vrf-export spoke;

```

```

Instance Routing Protocol  protocols {
                           ospf {
                             export redistribute-vpn;
                             area 0.0.0.0 {
                               interface fe-1/0/1.0;
                             }
                           }
                         }
}

```

```

Routing Options (Master Instance) routing-options {
                                   autonomous-system 1 loops 1;
                                   }

```

```

Protocols (Master Instance) protocols {

```

```

  Enable LDP  ldp {
              interface fe-1/0/0.0;
              }

```

Configure IBGP

```

bgp {
  group Spoke-F-to-Hub {
    type internal;
    local-address 10.255.14.182;
    neighbor 10.255.14.174 {
      family inet-vpn {
        unicast;
      }
    }
  }
}

```

Configure VPN Policy

```

policy-options {
  policy-statement hub {
    term a {
      from {
        protocol bgp;
        community hub;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement spoke {
    term a {
      from protocol ospf;
      then {
        community add spoke;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  policy-statement redistribute-vpn {
    term a {
      from {
        protocol bgp;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  community hub members target:65535:1;
  community spoke members target:65535:2;
}

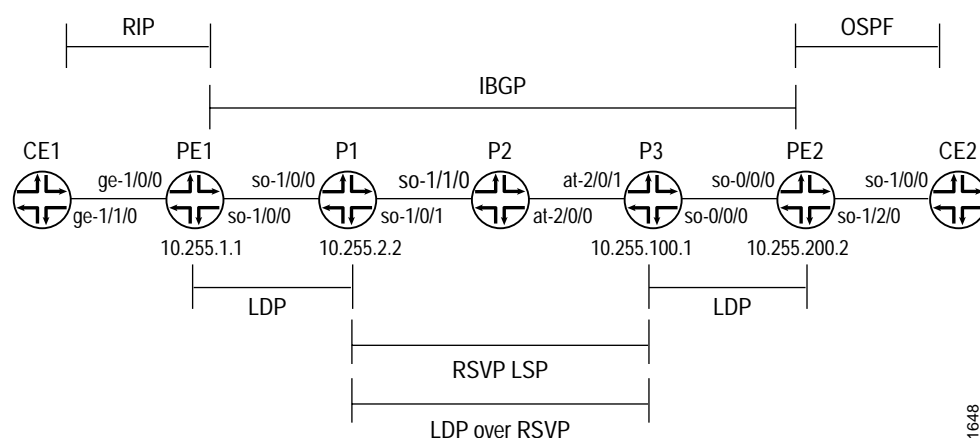
```

Configure an LDP-over-RSVP VPN Topology

This example shows how to set up a VPN topology in which LDP packets are tunneled over an RSVP LSP. This configuration consists of the following components (see Figure 19):

- One VPN (VPN-A)
- Two PE routers
- LDP as the signaling protocol between the PE routers and their adjacent provider routers
- An RSVP LSP between two of the provider routers over which LDP is tunneled

Figure 19: Example of an LDP-over-RSVP VPN Topology



The following steps describe how this topology is established and how packets are sent from CE Router CE2 to CE Router CE1:

1. The provider routers P1 and P3 establish RSVP LSPs between each other and install their loopback addresses in their inet.3 routing tables.
2. PE Router PE1 establishes an LDP session with Router P1 over interface so-1/0/0.0.
3. Router P1 establishes an LDP session with Router P3's loopback address, which is reachable using the RSVP LSP.
4. Router P1 sends its label bindings, which include a label to reach Router PE1, to Router P3. These label bindings allow Router P3 to direct LDP packets to Router PE1.
5. Router P3 establishes an LDP session with Router PE2 over interface so0-0/0/0.0 and establishes an LDP session with Router P1's loopback address.
6. Router P3 sends its label bindings, which include a label to reach Router PE2, to Router P1. These label bindings allow Router P1 to direct LDP packets to Router PE2's loopback address.
7. Routers PE1 and PE2 establish IBGP sessions with each other.

8. When Router PE1 announces to Router PE2 routes that it learned from Router CE1, it includes its VPN label. (The PE router creates the VPN label and binds it to the interface between the PE and CE routers.) Similarly, when Router PE2 announces routes that it learned from Router CE2, it sends its VPN label to Router PE1.

When Router PE2 wants to forward a packet to Router CE1, it pushes two labels onto the packet's label stack: first the VPN label that is bound to the interface between Router PE1 and Router CE1, then the LDP label used to reach Router PE1. Then it forwards the packets to Router P3 over interface so-0/0/1.0.

9. When Router P3 receives the packets from Router PE2, it swaps the LDP label that is on top of the stack (according to its LDP database) and also pushes an RSVP label onto the top of the stack so that the packet can now be switched by the RSVP LSP. At this point, there are three labels on the stack: the inner (bottom) label is the VPN label, the middle is the LDP label, and the outer (top) is the RSVP label.
10. Router P2 receives the packet and switches it to Router P1 by swapping the RSVP label. In this topology, because Router P2 is the penultimate-hop router in the LSP, it pops the RSVP label and forwards the packet over interface so-1/1/0.0 to Router P1. At this point, there are two labels on the stack: the inner label is the VPN label and the outer one is the LDP label.
11. When Router P1 receives the packet, it pops the outer label (the LDP label) and forwards the packet to Router PE1 using interface so-1/0/0.0. In this topology, Router PE1 is the egress LDP router, so Router P1 pops the LDP label instead of swapping it with another label. At this point, there is only one label on the stack, the VPN label.
12. When Router PE1 receives the packet, it pops the VPN label and forwards the packet as an IPv4 packet to Router CE1 over interface ge-1/1/0.0.

A similar set of operations occurs for packets sent from Router CE1 that are destined for Router CE2.

The following list explains how, for packets being sent from Router CE2 to Router CE1, the LDP, RSVP, and VPN labels are announced by the various routers. These steps include examples of label values (illustrated in Figure 20).

■ LDP labels

- Router PE1 announces LDP label 3 for itself to Router P1.
- Router P1 announces LDP label 100,001 for Router PE1 to Router P3.
- Router P3 announces LDP label 100,002 for Router PE1 to Router PE2.

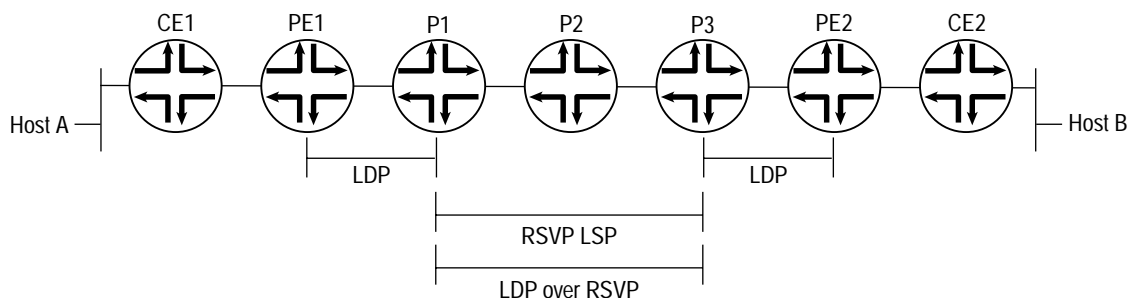
■ RSVP labels

- Router P1 announces RSVP label 3 to Router P2.
- Router P2 announces RSVP label 100,003 to Router P3.

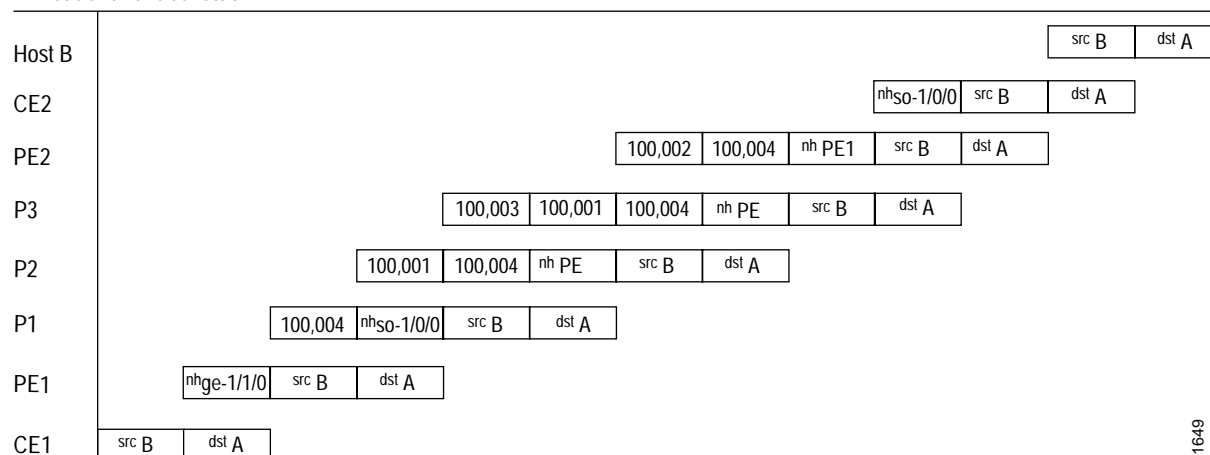
■ VPN label

- Router PE1 announces VPN label 100,004 to Router PE2 for the route from Router CE1 to Router CE2.

Figure 20: Label Pushing and Popping



IP header and label stack



1649

For a packet sent from Host B in Figure 20 to Host A, the packet headers and labels change as follows as the packet travels to its destination:

1. The packet that originates from Host B has a source address of B and a destination address of A in its header.
2. Router CE2 adds to the packet a next-hop of interface so-1/0/0.
3. Router PE2 swaps out the next-hop of interface so-1/0/0 and replaces it with a next-hop of PE1. It also adds two labels for reaching Router PE1, first the VPN label (100,004), then the LDP label (100,002). The VPN label is thus the inner (bottom) label on the stack, and the LDP label is the outer label.
4. Router P3 swaps out the LDP label added by Router PE2 (100,002) and replaces it with its LDP label for reaching Router PE1 (100,001). It also adds the RSVP label for reaching Router P2 (100,003).
5. Router P2 removes the RSVP label (100,003) because it is the penultimate hop in the MPLS LSP.
6. Router P1 removes the LDP label (100,001) because it is the penultimate LDP router. It also swaps out the next-hop of PE1 and replaces it with the next-hop interface, so-1/0/0.

7. Router PE1 removes the VPN label (100,004). It also swaps out the next-hop interface of so-1/0/0 and replaces it with its next-hop interface, ge-1/1/0.
8. Router CE1 removes the next-hop interface of ge-1/1/0, and the packet header now contains just a source address of B and a destination address of A.

The following sections explain how to configure the VPN functionality on the PE and provider routers. The CE routers are not aware of the VPN, so you configure them normally.

- Enable an IGP on the PE and Provider Routers on page 161
- Enable LDP on the PE and Provider Routers on page 162
- Enable RSVP and MPLS on the Provider Router on page 163
- Configure the MPLS LSP Tunnel between the Provider Routers on page 163
- Configure IBGP on the PE Routers on page 164
- Configure Routing Instances for VPNs on the PE Routers on page 165
- Configure VPN Policy on the PE Routers on page 166

The final section in this example, “LDP-over-MPLS VPN Configuration Summarized by Router” on page 168, consolidates the statements needed to configure VPN functionality on each of the service provider routers shown in Figure 19.



Note

In this example, a private AS number is used for the route distinguisher and the route target. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

Enable an IGP on the PE and Provider Routers

To allow the PE and provider routers to exchange routing information among themselves, you must configure an IGP on all these routers or you must configure static routes. You configure the IGP on the master instance of the routing protocol process (rpd) (that is, at the [edit protocols] hierarchy level), not within the VPN routing instance (that is, not at the [edit routing-instances] hierarchy level).

You configure the IGP in the standard way. This configuration example does not include this portion of the configuration.

Enable LDP on the PE and Provider Routers

In this configuration example, the LDP is the signaling protocol between the PE routers. For the VPN to function, you must configure LDP on the two PE routers and on the provider routers that are connected to the PE routers. You need to configure LDP only on the interfaces in the core of the service provider's network; that is, between the PE and provider routers and between the provider routers. You do not need to configure LDP on the interface between the PE and CE routers.

In this configuration example, you configure LDP on the provider routers' loopback interfaces because these are the interfaces on which the MPLS LSP is configured.

On the PE routers, you must also configure family inet when you configure the logical interface.

On Router PE1, configure LDP as follows:

```
[edit protocols]
ldp {
  interface so-1/0/0.0;
}
[edit interfaces]
so-1/0/0 {
  unit 0 {
    family mpls;
  }
}
```

On Router PE2, configure LDP as follows:

```
[edit protocols]
ldp {
  interface so-0/0/0.0;
}
[edit interfaces]
so-0/0/1 {
  unit 0 {
    family mpls;
  }
}
```

On Router P1, configure LDP as follows:

```
[edit protocols]
ldp {
  interface so-1/0/0.0;
  interface lo0;
}
```

On Router P3, configure LDP as follows:

```
[edit protocols]
ldp {
  interface lo0;
  interface so-0/0/0.0;
}
```

On Router P2, although you do not need to configure LDP, you can optionally configure it to provide a fallback LDP path in case the RSVP LSP becomes nonoperational:

```
[edit protocols]
ldp {
  interface so-1/1/0.0;
  interface at-2/0/0.0;
}
```

Enable RSVP and MPLS on the Provider Router

On the provider router, P2, you must configure RSVP and MPLS because this router exists on the MPLS LSP path between the provider Routers P1 and P3:

```
[edit]
protocols {
  rsvp {
    interface so-1/1/0.0;
    interface at-2/0/0.0;
  }
  mpls {
    interface so-1/1/0.0;
    interface at-2/0/0.0;
  }
}
```

Configure the MPLS LSP Tunnel between the Provider Routers

In this configuration example, LDP is tunneled over an RSVP LSP. Therefore, in addition to configuring RSVP, you must enable traffic engineering support in an IGP and you must create an MPLS LSP to tunnel the LDP traffic.

On Router P1, enable RSVP and configure one end of the MPLS LSP tunnel. In this example, traffic engineering support is enabled for OSPF, and you configure MPLS on the interfaces to the LSP and to Router PE1. In the to statement, you specify the loopback address of Router P3.

```
[edit]
protocols {
  rsvp {
    interface so-1/0/1.0;
  }
  mpls {
    label-switched-path P1-to-P3 {
      to 10.255.100.1;
      ldp-tunneling;
    }
    interface so-1/0/0.0;
    interface so-1/0/1.0;
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface so-1/0/0.0;
      interface so-1/0/1.0;
    }
  }
}
```

On Router P3, enable RSVP and configure the other end of the MPLS LSP tunnel. Again, traffic engineering support is enabled for OSPF, and you configure MPLS on the interfaces to the LSP and to Router PE2. In the to statement, you specify the loopback address of Router P1.

```
[edit]
protocols {
  rsvp {
    interface at-2/0/1.0;
  }
  mpls {
    label-switched-path P3-to-P1 {
      to 10.255.2.2;
      ldp-tunneling;
    }
    interface at-2/0/1.0;
    interface so-0/0/0.0;
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface at-2/0/1.0;
      interface so-0/0/0.0;
    }
  }
}
```

Configure IBGP on the PE Routers

On the PE routers, configure an IBGP session with the following properties:

- **VPN family**—To indicate that the IBGP session is for the VPN, include the family inet-vpn statement.
- **Loopback address**—Include the local-address statement, specifying the local PE router's loopback address. The IBGP session for VPNs runs through the loopback address. Note that you must also configure the lo0 interface at the [edit interfaces] hierarchy level. The example does not include this part of the router's configuration.
- **Neighbor address**—Include the neighbor statement, specifying the IP address of the neighboring PE router, which is its loopback (lo0) address.

On Router PE1, configure IBGP as follows:

```
[edit]
protocols {
  bgp {
    group PE1-to-PE2 {
      type internal;
      local-address 10.255.1.1;
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.200.2;
    }
  }
}
```

On Router PE2, configure IBGP as follows:

```
[edit]
protocols {
  bgp {
    group PE2-to-PE1 {
      type internal;
      local-address 10.255.200.2;
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.1.1;
    }
  }
}
```

Configure Routing Instances for VPNs on the PE Routers

Both PE routers service VPN-A, so you must configure one routing instance on each router for the VPN in which you define the following:

- Route distinguisher, which must be unique for each routing instance on the PE router. It is used to distinguish the addresses in one VPN from those in another VPN.
- Instance type of `vrf`, which creates the VRF table on the PE router.
- Interfaces connected to the CE routers.
- VRF import and export policies, which must be the same on each PE router that services the same VPN. Unless the import policy contains only a `then reject` statement, it must include reference to a community. Otherwise, when you try to commit the configuration, the commit fails.



Note

In this example, a private AS number is used for the route distinguisher. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

- Routing between the PE and CE routers, which is required for the PE router to distribute VPN-related routes to and from connected CE routers. You can configure a routing protocol—BGP, OSPF, or RIP—or you can configure static routing.

On Router PE1, configure the following routing instance for VPN-A. In this example, Router PE1 uses RIP to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
  VPN-A {
    instance-type vrf;
    interface ge-1/0/0.0;
    route-distinguisher 65535:0;
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
    protocols {
      rip {
        group PE1-to-CE1 {
          neighbor ge-1/0/0.0;
        }
      }
    }
  }
}
```

On Router PE2, configure the following routing instance for VPN-A. In this example, Router PE2 uses OSPF to distribute routes to and from the CE router to which it is connected.

```
[edit]
routing-instance {
  VPN-A {
    instance-type vrf;
    interface so-1/2/0.0;
    route-distinguisher 65535:1;
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
    protocols {
      ospf {
        area 0.0.0.0 {
          interface so-1/2/0.0;
        }
      }
    }
  }
}
```

Configure VPN Policy on the PE Routers

You must configure VPN import and export policies on each of the PE routers so that they install the appropriate routes in their VRF tables, which they use to forward packets within a VPN. For VPN-A, the VRF table is VPN-A.inet.0.

In the VPN policy, you also configure VPN target communities.



Note

In this example, a private AS number is used for the route target. This number is used for illustration only. When you are configuring VPNs, you should use an assigned AS number.

On Router PE1, configure the following VPN import and export policies.



Note

The policy qualifiers shown in this example are only those needed for the VPN to function. You can configure additional qualifiers, as needed, to any policies that you configure.

```
[edit]
policy-options {
  policy-statement VPN-A-import {
    term a {
      from {
        protocol bgp;
        community VPN-A;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement VPN-A-export {
    term a {
      from protocol rip;
      then {
        community add VPN-A;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community VPN-A members target:65535:00;
}
```

On Router PE2, configure the following VPN import and export policies:

```
[edit]
policy-options {
  policy-statement VPN-A-import {
    term a {
      from {
        protocol bgp;
        community VPN-A;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
}
```

```

policy-statement VPN-A-export {
  term a {
    from protocol ospf;
    then {
      community add VPN-A;
      accept;
    }
  }
  term b {
    then reject;
  }
}
community VPN-A members target:65535:00;
}

```

To apply the VPN policies on the routers, include the vrf-export and vrf-import statements when you configure the routing instance on the PE routers. The VRF import and export policies handle the route distribution across the IBGP session running between the PE routers.

LDP-over-MPLS VPN Configuration Summarized by Router

Router PE1

```

Routing Instance for VPN-A  routing-instance {
                             VPN-A {
                               instance-type vrf;
                               interface ge-1/0/0.0;
                               route-distinguisher 65535:0;
                               vrf-import VPN-A-import;
                               vrf-export VPN-A-export;
                             }
                           }

Instance Routing Protocol    protocols {
                             rip {
                               group PE1-to-CE1 {
                                 neighbor ge-1/0/0.0;
                               }
                             }
                           }

Interfaces                  interfaces {
                             so-1/0/0 {
                               unit 0 {
                                 family mpls;
                               }
                             }
                             ge-1/0/0 {
                               unit 0;
                             }
                           }

Master Protocol Instance    protocols {

Enable LDP                  ldp {
                             interface so-1/0/0.0;
                           }
}

```



```

Enable MPLS      mpls {
                    interface so-1/0/0.0;
                    interface ge-1/0/0.0;
                    }

Configure IBGP   bgp {
                    group PE1-to-PE2 {
                        type internal;
                        local-address 10.255.1.1;
                        family inet-vpn {
                            unicast;
                        }
                    }
                    neighbor 10.255.100.1;
                }

Configure VPN Policy policy-options {
                    policy-statement VPN-A-import {
                        term a {
                            from {
                                protocol bgp;
                                community VPN-A;
                            }
                            then accept;
                        }
                        term b {
                            then reject;
                        }
                    }
                    policy-statement VPN-A-export {
                        term a {
                            from protocol rip;
                            then {
                                community add VPN-A;
                                accept;
                            }
                        }
                        term b {
                            then reject;
                        }
                    }
                    community VPN-A members target:65535:00;
                }

```

Router P1

```

Master Protocol Instance protocols {

    Enable RSVP      rsvp {
                        interface so-1/0/1.0;
                    }

    Enable LDP       ldp {
                        interface so-1/0/0.0;
                        interface lo0.0;
                    }

```

```

Enable MPLS      mpls {
                  label-switched-path P1-to-P3 {
                    to 10.255.100.1;
                    ldp-tunneling;
                  }
                  interface so-1/0/0.0;
                  interface so-1/0/1.0;
                }

```

```

Configure OSPF for Traffic Engineering Support  ospf {
                                                  traffic-engineering;
                                                  area 0.0.0.0 {
                                                    interface so-1/0/0.0;
                                                    interface so-1/0/1.0;
                                                  }
                                                  }

```

Router P2

```

Master Protocol Instance  protocols {

Enable RSVP              rsvp {
                          interface so-1/1/0.0;
                          interface at-2/0/0.0;
                        }

Enable MPLS              mpls {
                          interface so-1/1/0.0;
                          interface at-2/0/0.0;
                        }

```

Router P3

```

Master Protocol Instance  protocols {

Enable RSVP              rsvp {
                          interface at-2/0/1.0;
                        }

Enable LDP               ldp {
                          interface so-0/0/0.0;
                          interface lo0.0;
                        }

Enable MPLS              mpls {
                          label-switched-path P3-to-P1 {
                            to 10.255.2.2;
                            ldp-tunneling;
                          }
                          interface at-2/0/1.0;
                          interface so-0/0/0.0;
                        }

```

Configure OSPF for Traffic Engineering Support

```
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface at-2/0/1.0;
    interface at-2/0/1.0;
  }
}
```

Router PE2**Routing Instance for VPN-A**

```
routing-instance {
  VPN-A {
    instance-type vrf;
    interface so-1/2/0.0;
    route-distinguisher 65535:1;
    vrf-import VPN-A-import;
    vrf-export VPN-A-export;
  }
}
```

Instance Routing Protocol

```
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-1/2/0.0;
    }
  }
}
```

Interfaces

```
interfaces {
  so-0/0/0 {
    unit 0 {
      family mpls;
    }
  }
  so-1/2/0 {
    unit 0;
  }
}
```

Master Protocol Instance

```
protocols {
```

Enable LDP

```
ldp {
  interface so-0/0/0.0;
}
```

Enable MPLS

```
mpls {
  interface so-0/0/0.0;
  interface so-1/2/0.0;
}
```

```

Configure IBGP      bgp {
                        group PE2-to-PE1 {
                          type internal;
                          local-address 10.255.200.2;
                          family inet-vpn {
                            unicast;
                          }
                          neighbor 10.255.1.1;
                        }
                      }
Configure VPN Policy policy-options {
                        policy-statement VPN-A-import {
                          term a {
                            from {
                              protocol bgp;
                              community VPN-A;
                            }
                            then accept;
                          }
                          term b {
                            then reject;
                          }
                        }
                        policy-statement VPN-A-export {
                          term a {
                            from protocol ospf;
                            then {
                              community add VPN-A;
                              accept;
                            }
                          }
                          term b {
                            then reject;
                          }
                        }
                        community VPN-A members target:65535:01;
                      }

```

Configure an Application-Based Layer 3 VPN Topology

This example illustrates an application-based mechanism for forwarding traffic into a Layer 3 VPN. Typically, one or more interfaces are associated with, or bound to, a VPN by including them in the configuration of the VPN routing instance. By binding the interface to the VPN, the VPN's VRF table is used to make forwarding decisions for any incoming traffic on that interface. Binding the interface also includes the interface local routes in the VRF, which provides next-hop resolution for VRF routes.

In this example, a firewall filter is used to define which incoming traffic on an interface is forwarded using the standard routing table, `inet.0`, and which incoming traffic is forwarded using the VRF table. You can expand this example such that incoming traffic on an interface can be redirected to one or more VPNs. For example, you can define a configuration to support a VPN that forwards traffic based on source address, that forwards Hypertext Transfer Protocol (HTTP) traffic, or that forwards only streaming media.

For this configuration to work, the following must be true:

- The interfaces that use filter-based forwarding must not be bound to the VPN.
- Static routing must be used as the means of routing.
- You must define an interface routing table group that is shared among inet.0 and the VRFs to provide local routes to the VRF.

This example consists of two client hosts (Client D and Client E) that are in two different VPNs and that want to send traffic both within the VPN and to the Internet. The paths are defined as follows:

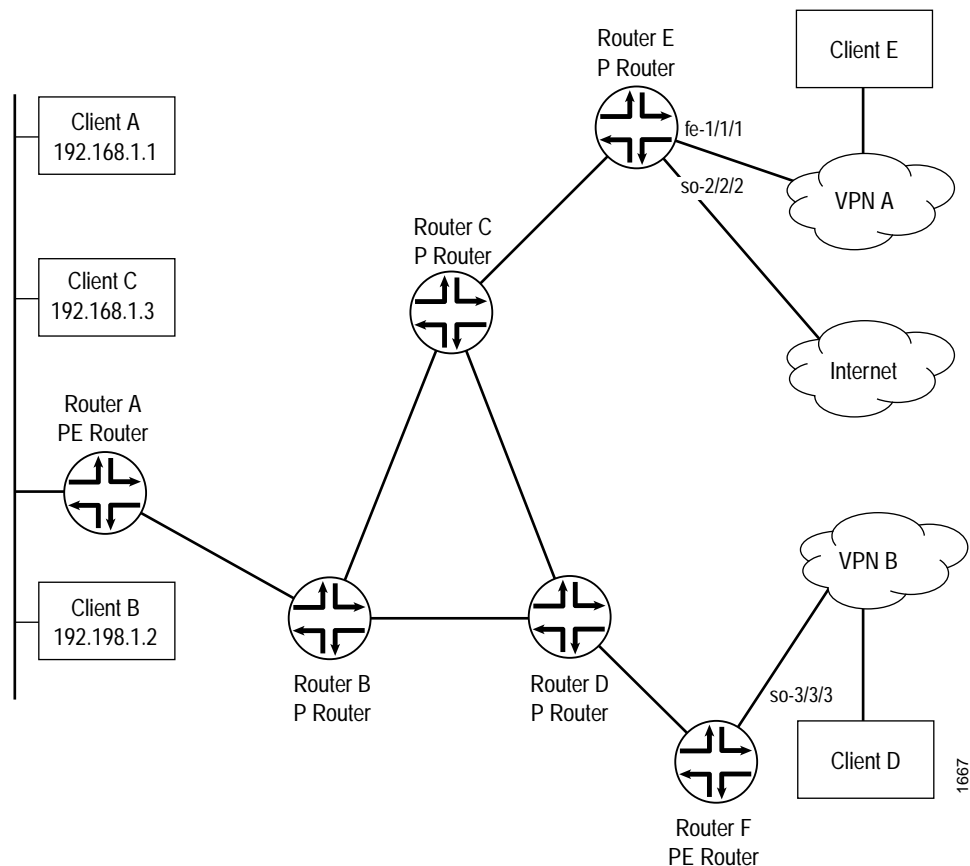
- Client A sends traffic to Client E over VPN A with a return path that also uses VPN A (using the VPN's VRF table).
- Client B sends traffic to Client D over VPN B with a return path that uses standard destination-based routing (using the inet.0 routing table).
- Clients B and C send traffic to the Internet using standard routing (using the inet.0 routing table), with a return path that also uses standard routing.

This example illustrates that there are a large variety of options in configuring an application-based Layer 3 VPN topology. This flexibility has application in many network implementations requiring specific traffic to be forwarded in a constrained routing environment.

This configuration example shows only the portions of the configuration for the filter-based forwarding, routing instances, and policy. It does not illustrate how to configure a Layer 3 VPN.

Figure 21 illustrates the configuration used in this example.

Figure 21: Application-Based Layer 3 VPN Example Configuration



Configuration on Router A

On Router A, you configure the interface to Clients A, B, and C. The configuration evaluates incoming traffic to determine whether it is to be forwarded using the VPN or using standard destination-based routing.

First, you apply an inbound filter and configure the interface.

```
[edit]
interfaces {
  fe-1/1/0 {
    unit 0 {
      family inet {
        filter {
          input fbv-vrf;
        }
        address 192.168.1.1/24;
      }
    }
  }
}
```

Because the interfaces that use filter-based forwarding must not be bound to a VPN, you must configure an alternate method to provide next-hop routes to the VRF table. You do this by defining an interface routing table group and sharing this group among all the routing tables. To provide a route back to the clients for normal inet.0 routing, you define a static route to include in inet.0 and redistribute the static route into BGP.

```
[edit]
routing-options {
  interface-routes {
    rib-group inet if-rib;
  }
  static {
    route 192.168.1.0/24 next-hop fe-1/1/0.0
  }
  rib-groups {
    if-rib {
      import-rib [ inet.0 vpn-A.inet.0 vpn-B.inet.0 ];
    }
  }
}
```

You apply the following filter to incoming traffic on interface fe-1/1/0.0. The first term matches traffic from Client A and forwards it to the routing instance for VPN A. The second term matches traffic from Client B that is destined for Client D and forwards it to the routing instance for VPN B. The third term matches all other traffic, which is forwarded normally using destination-based forwarding according to the routes in inet.0.

```
[edit firewall family family-name]
filter fbf-vrf {
  term vpnA {
    from {
      source-address {
        192.168.1.1/32;
      }
    }
    then {
      routing-instance vpn-A;
    }
  }
  term vpnB {
    from {
      source-address {
        192.168.1.2/32;
      }
      destination-address {
        192.168.3.0/24;
      }
    }
    then routing-instance vpn-B;
  }
  term internet {
    then accept;
  }
}
```

You then configure the routing instances for VPN A and VPN B. Notice that these statements include all the required statements to define a Layer 3 VPN except for the interface statement.

```
[edit]
routing-instances {
  vpn-A {
    instance-type vrf;
    route-distinguisher 172.21.10.63:100;
    vrf-import vpn-A-import;
    vrf-export vpn-A-export;
    routing-options {
      static {
        route 192.168.1.0/24 next-hop fe-1/1/0.0;
      }
    }
  }
  vpn-B {
    instance-type vrf;
    route-distinguisher 172.21.10.63:200;
    vrf-import vpn-B-import;
    vrf-export vpn-B-export;
    routing-options {
      static {
        route 192.168.1.0/24 next-hop fe-1/1/0.0;
      }
    }
  }
}
```

Configuration on Router E

On Router E, you configure a default route to reach the Internet. You should inject this route into the local IGP mesh to provide an exit point from the network.

```
[edit]
routing-options {
  static {
    route 0.0.0.0/0 next-hop so-2/2/2.0 discard
  }
}
```

You configure the interface to Client E so that all incoming traffic on interface fe-1/1/1.0 that matches the VPN policy is forwarded over VPN A:

```
[edit]
routing-instances {
  vpn-A {
    interface fe-1/1/1.0
    instance-type vrf;
    route-distinguisher 172.21.10.62:100;
    vrf-import vpn-A-import;
    vrf-export vpn-A-export;
    routing-options {
      static {
        route 192.168.2.0/24 next-hop fe-1/1/1.0;
      }
    }
  }
}
```


Configuration for Router F

Again, because the interfaces that use filter-based forwarding must not be bound to a VPN, you configure an alternate method to provide next-hop routes to the VRF table by defining an interface routing table group and sharing this group among all the routing tables. To provide a route back to the clients for normal inet.0 routing, you define a static route to include in inet.0 and redistribute the static route into BGP.

```
[edit]
routing-options {
  interface-routes {
    rib-group inet if-rib;
  }
  rib-groups {
    if-rib {
      import-rib [ inet.0 vpn-B.inet.0];
    }
  }
}
```

To direct traffic from VPN B to Client D, you configure the routing instance for VPN B on Router F. All incoming traffic from Client D on interface so-3/3/3.0 is forwarded normally using the destination address based on the routes in inet.0.

```
[edit]
routing-instances {
  vpn-B {
    instance-type vrf;
    route-distinguisher 172.21.10.64:200;
    vrf-import vpn-B-import;
    vrf-export vpn-B-export;
    routing-options {
      static {
        route 192.168.3.0/24 next-hop so-3/3/3.0;
      }
    }
  }
}
```

Configure an OSPF Domain ID for a Layer 3 VPN

This example illustrates how to configure an OSPF domain ID for a VPN using OSPF as the routing protocol between the PE and CE routers. Routes from an OSPF domain need an OSPF domain ID when they are distributed in BGP as VPN-IPv4 routes in VPNs with multiple OSPF domains. In a VPN connecting multiple OSPF domains, the routes from one domain might overlap with the routes of another.

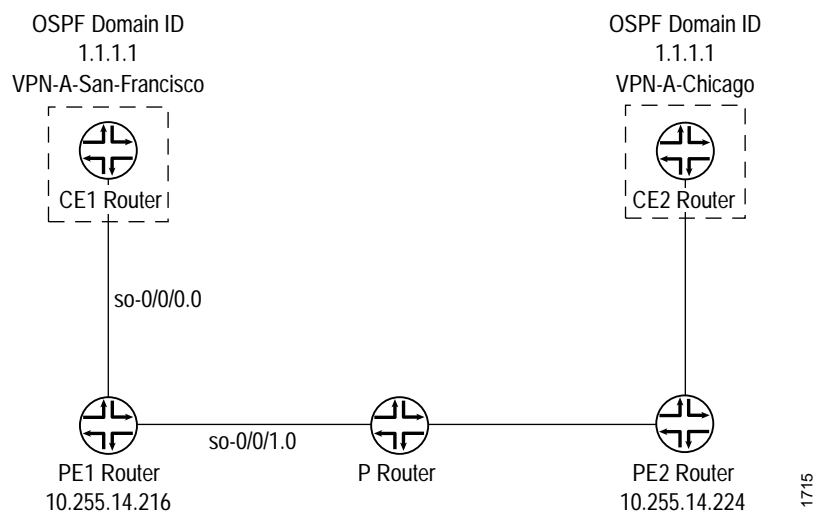
Configuring a unique OSPF domain ID for each domain ensures that the routes for each domain remain separate. If a domain ID is not configured, the default value is 0.0.0.0. In addition, if the remote PE router does not advertise a domain ID in the VPN-IPv4 routes, the local PE router assumes the domain ID matches the remote PE routers, and an OSPF Type 3 LSA is issued for the routes. Each VRF table in a PE router associated with an OSPF instance is configured with the same OSPF domain ID.

Whether a route is redistributed and advertised as a Type 3 LSA or as a Type 5 LSA depends on the following:

- If the receiving PE router sees a Type 3 route with a matching domain ID, the route is redistributed and advertised as a Type 3 LSA.
- If the receiving PE router sees a Type 5 route with a matching domain ID, the route is redistributed and advertised as a Type 5 LSA.
- If the receiving PE router sees a Type 3 route without a domain ID, the route is redistributed and advertised as a Type 3 LSA.
- If the receiving PE router sees a Type 3 route with a non-matching domain ID, the route is redistributed and advertised as a Type 5 LSA.
- If the receiving PE router sees a Type 5 route with a non-matching domain ID, the route is redistributed and advertised as a Type 5 LSA.

Figure 22 shows this example's configuration topology. Only the configuration for Router PE1 is provided. The configuration for Router PE2 can be similar to the configuration for Router PE1. There are no special configuration requirements for the CE routers.

Figure 22: Example of a Configuration Using an OSPF Domain ID



Configure Interfaces on Router PE1

You need to configure two interfaces for Router PE1—the so-0/0/0 interface for traffic to Router CE1 (San Francisco) and the so-0/0/1 interface for traffic to a provider (P) router in the service provider's network.

Configure the interfaces for Router PE1:

```
[edit]
interfaces {
  so-0/0/0 {
    unit 0 {
      family inet {
        address 10.19.1.2/30;
      }
    }
  }
  so-0/0/1 {
    unit 0 {
      family inet {
        address 10.19.2.1/30;
      }
      family mpls;
    }
  }
}
```

Configure Routing Options on Router PE1

At the [edit routing-options] hierarchy level, you need to configure the router-id and autonomous-system statements. The router-id statement identifies Router PE1.

Configure the routing options for Router PE1:

```
[edit]
routing-options {
  router-id 10.255.14.216;
  autonomous-system 69;
}
```

Configure Protocols on Router PE1

On Router PE1, you need to configure MPLS, BGP, OSPF, and LDP at the [edit protocols] hierarchy level:

```
[edit]
protocols {
  mpls {
    interface so-0/0/0.0;
  }
  bgp {
    group San-Francisco-Chicago {
      type internal;
      preference 10;
      local-address 10.255.14.216;
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.14.224;
    }
  }
  ospf {
    traffic-engineering;
    area 0.0.0.0 {
      interface so-0/0/1.0;
    }
  }
  ldp {
    interface so-0/0/1.0;
  }
}
```

Configure Policy Options on Router PE1

On Router PE1, you need to configure policies at the [edit policy-options] hierarchy level. These policies ensure that the CE routers in the Layer 3 VPN exchange routing information. In this example, Router CE1 in San Francisco exchanges routing information with Router CE2 in Chicago.

Configure the policy options on the PE1 router:

```
[edit]
policy-options {
  policy-statement vpn-import-VPN-A {
    term term1 {
      from {
        protocol bgp;
        community import-target-VPN-A;
      }
      then accept;
    }
    term term2 {
      then reject;
    }
  }
}
```

```

policy-statement vpn-export-VPN-A {
  term term1 {
    from protocol ospf;
    then {
      community add export-target-VPN-A;
      accept;
    }
  }
  term term2 {
    then reject;
  }
}
community export-target-VPN-B members [ target:10.255.14.216:11 domain-id:1.1.1.1:0 ];
community import-target-VPN-B members target:10.255.14.224:31;
}

```

Configure the Routing Instance on Router PE1

You need to configure a Layer 3 VPN routing instance on Router PE1. To indicate that the routing instance is for a Layer 3 VPN, add the instance-type vrf statement at the [edit routing-instance *routing-instance-name*] hierarchy level.

The domain-id statement is configured at the [edit routing-instances routing-options protocols ospf] hierarchy level. As shown in Figure 22 on page 178, the routing instance on Router PE2 must share the same domain ID as the corresponding routing instance on Router PE1 so that routes from Router CE1 to Router CE2 and vice versa are distributed as Type 3 LSAs. If you configure different OSPF domain IDs in the routing instances for Router PE1 and Router PE2, the routes from each CE router will be distributed as Type 5 LSAs.

Configure the routing instance on Router PE1:

```

[edit]
routing-instances {
  VPN-A-San-Francisco-Chicago {
    instance-type vrf;
    interface so-0/0/0.0;
    route-distinguisher 10.255.14.216:11;
    vrf-import vpn-import-VPN-A;
    vrf-export vpn-export-VPN-A;
    routing-options {
      router-id 10.255.14.216;
      autonomous-system 69;
    }
    protocols {
      ospf {
        domain-id 1.1.1.1;
        export vpn-import-VPN-A;
        area 0.0.0.0 {
          interface so-0/0/0.0;
        }
      }
    }
  }
}
}

```

Configuration Summary for Router PE1

```

Configure Interfaces interfaces {
    so-0/0/0 {
        unit 0 {
            family inet {
                address 10.19.1.2/30;
            }
        }
    }
    so-0/0/1 {
        unit 0 {
            family inet {
                address 10.19.2.1/30;
            }
            family mpls;
        }
    }
}

Configure Routing Options routing-options {
    router-id 10.255.14.216;
    autonomous-system 69;
}

Configure Protocols protocols {
    mpls {
        interface so-0/0/0.0;
    }
    bgp {
        group San-Francisco-Chicago {
            type internal;
            preference 10;
            local-address 10.255.14.216;
            family inet-vpn {
                unicast;
            }
            neighbor 10.255.14.224;
        }
    }
    ospf {
        traffic-engineering;
        area 0.0.0.0 {
            interface so-0/0/1.0;
        }
    }
    ldp {
        interface so-0/0/1.0;
    }
}

Configure VPN Policy policy-options {
    policy-statement vpn-import-VPN-A {
        term term1 {
            from {
                protocol bgp;
                community import-target-VPN-A;
            }
            then accept;
        }
    }
}

```

```

        term term2 {
            then reject;
        }
    }
    policy-statement vpn-export-VPN-A {
        term term1 {
            from protocol ospf;
            then {
                community add export-target-VPN-A;
                accept;
            }
        }
        term term2 {
            then reject;
        }
    }
    community export-target-VPN-B members [ target:10.255.14.216:11 domain-id:1.1.1.1:0 ];
    community import-target-VPN-B members target:10.255.14.224:31;
}

```

Routing Instance for Layer 3 VPN

```

routing-instances {
    VPN-A-San-Francisco-Chicago {
        instance-type vrf;
        interface so-0/0/0.0;
        route-distinguisher 10.255.14.216:11;
        vrf-import vpn-import-VPN-A;
        vrf-export vpn-export-VPN-A;
        routing-options {
            router-id 10.255.14.216;
            autonomous-system 69;
        }
        protocols {
            ospf {
                domain-id 1.1.1.1;
                export vpn-import-VPN-A;
                area 0.0.0.0 {
                    interface so-0/0/0.0;
                }
            }
        }
    }
}

```

Configure Overlapping VPNs Using Routing Table Groups

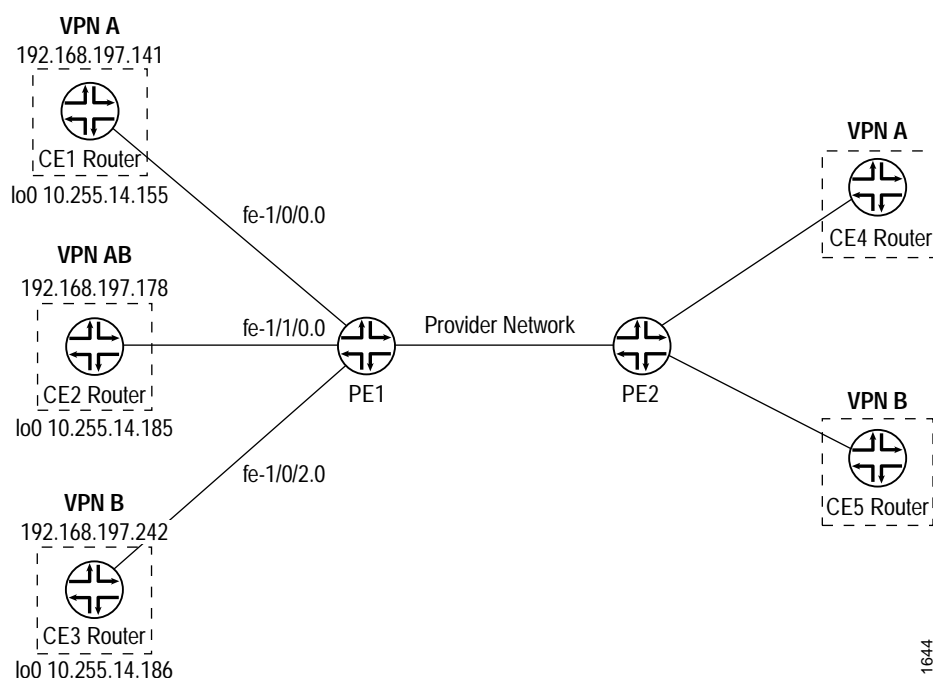
In Layer 3 VPNs, a CE router is often a member of more than one VPN. This example illustrates how to configure PE routers that support CE routers that support multiple VPNs. Support for this type of configuration uses a JUNOS software feature called routing table groups (sometimes also called routing information base [RIB] groups), which allows a route to be installed into several routing tables. A routing table group is a list of routing tables into which the protocol should install its routes.

You define routing table groups at the [edit routing-options] hierarchy level for the default instance. You cannot configure routing table groups at the [routing-instances routing-options] hierarchy level; doing so results in a commit error.

After you define a routing table group, it can be used by multiple protocols. You can also apply routing table groups to static routing. The configuration examples in this section include both types of configurations.

Figure 23 illustrates the topology for the configuration example in this section. The configurations in this section illustrate local connectivity between CE routers connected to the same PE router. If Router PE1 were connected only to Router CE2 (VPN AB), there would be no need for any extra configuration. The configuration statements in the sections that follow enable VPN AB Router CE2 to communicate with VPN A Router CE1 and VPN B Router CE3 that are directly connected to the Router PE1. VPN routes that originate from the remote PE routers (the PE2 router in this case) are placed in a global Layer 3 VPN routing table (bgp.l3vpn.inet.0) and routes with appropriate route targets are imported into the routing tables as dictated by the VRF import policy configuration. The goal is to be able to choose routes from individual VPN routing tables that are locally populated.

Figure 23: Example of an Overlapping VPN Topology



The following sections explain several ways to configure overlapping VPNs. For all the examples that follow, you need to configure routing table groups as described in “Configure Routing Table Groups” on page 185.

The following sections illustrate different scenarios for configuring overlapping VPNs, depending on the routing protocol used between the PE and CE routers. For all of these examples, you need to configure routing table groups.

- Configure Static Routes between the PE and CE Routers on page 185
- Configure BGP between the PE and CE Routers on page 190
- Configure OSPF between the PE and CE Routers on page 192
- Configure Static, BGP, and OSPF Routes between the PE and CE Routers on page 193

Router PE1 is where all the filtering and configuration modification takes place. Therefore only VPN configurations for PE1 are shown. The CE routers do not know the VPN exists, so you can configure them normally.

Configure Routing Table Groups

In this example, routing table groups are common in the four configuration scenarios. The routing table groups are used to install routes (including interface, static, OSPF, and BGP routes) into several routing tables for the default and other instances. In the routing table group definition, the first routing table is called the primary routing table. (Normally, the primary routing table is the table into which the route would be installed if you did not configure routing table groups. The other routing tables are called secondary routing tables.)

The routing table groups in this configuration install routes as follows:

- `vpna-vpnab` installs routes into routing tables `VPN-A.inet.0` and `VPN-AB.inet.0`.
- `vpnb-vpnab` installs routes into routing tables `VPN-B.inet.0` and `VPN-AB.inet.0`.
- `vpnab-vpna_and_vpnb` installs routes into routing tables `VPN-AB.inet.0`, `VPN-A.inet.0`, and `VPN-B.inet.0`.

Configure the routing table groups as follows:

```
[edit]
routing-options {
  rib-groups {
    vpna-vpnab {
      import-rib [ VPN-A.inet.0 VPN-AB.inet.0 ];
    }
    vpnb-vpnab {
      import-rib [ VPN-B.inet.0 VPN-AB.inet.0 ];
    }
    vpnab-vpna_and_vpnb {
      import-rib [ VPN-AB.inet.0 VPN-A.inet.0 VPN-B.inet.0 ];
    }
  }
}
```

Configure Static Routes between the PE and CE Routers

To configure static routing between the PE1 router and the CE1, CE2, and CE3 routers, you must configure routing instances for VPN A, VPN B, and VPN AB (you configure static routing under each instance):

- Configure the Routing Instance for VPN A on page 186
- Configure the Routing Instance for VPN AB on page 186
- Configure the Routing Instance for VPN B on page 187
- Configure VPN Policy on page 188

Configure the Routing Instance for VPN A

On Router PE1, configure VPN A as follows:

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface fe-1/0/0.0;
    route-distinguisher 10.255.14.175:3;
    vrf-import vpn-a-import;
    vrf-export vpn-a-export;
    routing-options {
      interface-routes {
        rib-group inet vpn-a-vpnab;
      }
      static {
        route 10.255.14.155/32 next-hop 192.168.197.141;
        route 10.255.14.185/32 next-hop 192.168.197.178;
      }
    }
  }
}
```

The interface-routes statement installs VPN A's interface routes into the routing tables defined in the routing table group vpn-a-vpnab.

The static statement configures the static routes that are installed in the VPN-A.inet.0 routing table. The first static route is for Router CE1 (VPN A) and the second is for Router CE2 (in VPN AB).

Note that next-hop 192.168.197.178 is not in VPN A. Route 10.255.14.185/32 cannot be installed in VPN-A.inet.0 unless interface routes from routing instance VPN AB are installed in this routing table. Including the interface-routes statements in the VPN AB configuration provides this next hop. Similarly, including the interface-routes statement in the VPN AB configuration installs 192.168.197.141 into VPN-AB.inet.0.

Configure the Routing Instance for VPN AB

On Router PE1, configure VPN AB as follows:

```
[edit]
routing instances {
  VPN-AB {
    instance-type vrf;
    interface fe-1/1/0.0;
    route-distinguisher 10.255.14.175:9;
    vrf-import vpnab-import;
    vrf-export vpnab-export;
    routing-options {
      interface-routes {
        rib-group vpnab-vpna_and_vpnab;
      }
    }
  }
}
```

```

static {
    route 10.255.14.185/32 next-hop 192.168.197.178;
    route 10.255.14.155/32 next-hop 192.168.197.141;
    route 10.255.14.186/32 next-hop 192.168.197.242;
}
}
}
}

```

In this configuration, the following static routes are installed in the VPN-AB.inet.0 routing table:

- 10.255.14.185/32 is for Router CE2 (in VPN AB)
- 10.255.14.155/32 is for Router CE1 (in VPN A)
- 10.255.14.186/32 is for Router CE3 (in VPN B)

192.168.197.141 and 192.168.197.242 do not belong to VPN AB. Routes 10.255.14.155/32 and 10.255.14.186/32 cannot be installed in VPN-AB.inet.0 unless interface routes from VPN A and VPN B are installed in this routing table. The interface route configurations in VPN A and VPN B routing instances provide these next hops.

Configure the Routing Instance for VPN B

On Router PE1, configure VPN B as follows:

```

[edit]
routing instances {
    VPN-B {
        instance-type vrf;
        interface fe-1/0/2.0;
        route-distinguisher 10.255.14.175:10;
        vrf-import vpnb-import;
        vrf-export vpnb-export;
        routing-options {
            interface-routes {
                rib-group inet vpnb-vpnab;
            }
            static {
                route 10.255.14.186/32 next-hop 192.168.197.242;
                route 10.255.14.185/32 next-hop 192.168.197.178;
            }
        }
    }
}

```

When you configure the routing instance for VPN B, these static routes are placed in VPNB.inet.0:

- 10.255.14.186/32 is for Router CE3 (in VPN B)
- 10.255.14.185/32 is for Router CE2 (in VPN AB)

192.168.197.178 does not belong to VPN B. Route 10.255.14.185/32 cannot be installed in VPN-B.inet.0 unless interface routes from VPN AB are installed in this routing table. The interface route configuration in VPN AB provides this next hop.

Configure VPN Policy

The vrf-import and vrf-export policy statements that you configure for overlapping VPNs are the same as policy statements for regular VPNs, except that you include the from interface statement in each VRF export policy. This statement forces each VPN to announce only those routes that originated from that VPN. For example, VPN A has routes that originated in VPN A and VPN AB. If you do not include the from interface statement, VPN A announces its own routes as well as VPN AB's routes, so the remote PE router receives multiple announcements for the same routes. Including the from interface statement restricts each VPN to announcing only the routes it originated and allows you to filter out the routes imported from other routing tables for local connectivity.

In this configuration example, the vpnab-import policy accepts routes from VPN A, VPN B, and VPN AB. The vpna-export policy only exports routes that originate in VPN A. Similarly, the vpnb-export and vpnab-export policies only export routes that originate within the respective VPNs.

On Router PE1, configure the following VPN import and export policies:

```
[edit]
policy-options {
  policy-statement vpna-import {
    term a {
      from {
        protocol bgp;
        community VPNA-comm;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement vpnb-import {
    term a {
      from {
        protocol bgp;
        community VPNB-comm;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement vpnab-import {
    term a {
      from {
        protocol bgp;
        community [ VPNA-comm VPNB-comm ];
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
}
```

```

policy-statement vpna-export {
  term a {
    from {
      protocol static;
      interface fe-1/0/0.0;
    }
    then {
      community add VPNA-comm;
      accept;
    }
  }
  term b {
    then reject;
  }
}
policy-statement vpnb-export {
  term a {
    from {
      protocol static;
      interface fe-1/0/2.0;
    }
    then {
      community add VPNB-comm;
      accept;
    }
  }
  term b {
    then reject;
  }
}
policy-statement vpnab-export {
  term a {
    from {
      protocol static;
      interface fe-1/1/0.0;
    }
    then {
      community add VPNB-comm;
      community add VPNA-comm;
      accept;
    }
  }
  term b {
    then reject;
  }
}
community VPNA-comm members target:69:1;
community VPNB-comm members target:69:2;
}

```

On Router PE1, apply the VPN import and export policies as follows:

```

[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface fe-1/0/0.0;
    route-distinguisher 10.255.14.175:3;
    vrf-import vpna-import;
    vrf-export vpna-export;
  }
}

```

```

routing-options {
  static {
    rib-group vpnab-vpnab;
    route 10.255.14.155/32 next-hop 192.168.197.141;
    route 10.255.14.185/32 next-hop 192.168.197.178;
  }
}
VPN-AB {
  instance-type vrf;
  interface fe-1/1/0.0;
  route-distinguisher 10.255.14.175:9;
  vrf-import vpnab-import;
  vrf-export vpnab-export;
  routing-options {
    static {
      rib-group vpnab-vpna_and_vpnab;
      route 10.255.14.185/32 next-hop 192.168.197.178;
    }
  }
}
VPN-B {
  instance-type vrf;
  interface fe-1/0/2.0;
  route-distinguisher 10.255.14.175:10;
  vrf-import vpnb-import;
  vrf-export vpnb-export;
  routing-options {
    static {
      rib-group vpnb-vpnab;
      route 10.255.14.186/32 next-hop 192.168.197.242;
    }
  }
}

```

For VPN A, include the routing-options statement at the [edit routing-instances *routing-instance-name*] hierarchy level to install the static routes directly into the routing tables defined in the routing table group vpnab-vpnab. For VPN AB, the configuration installs the static route directly into the routing tables defined in the routing table group vpnab-vpna and vpnab-vpnab. For VPN B the configuration installs the static route directly into the routing tables defined in the routing table group vpnb-vpnab.

Configure BGP between the PE and CE Routers

In this configuration example, the vpnab-site1 BGP group for VPN A installs the routes learned from the BGP session into the routing tables defined in the vpnab-vpnab routing table group. For VPN AB, the vpnab-site1 group installs the routes learned from the BGP session into the routing tables defined in the vpnab-vpna_and_vpnab routing table group. For VPN B, the vpnb-site1 group installs the routes learned from the BGP session into the routing tables defined in the vpnb-vpnab routing table group. Note that interface routes are not needed for this configuration.

The VRF import and export policies are similar to those defined in “Configure Static Routes between the PE and CE Routers” on page 185, except the export protocol is BGP instead of a static route. On all vrf-export policies, you use the from protocol bgp statement.

On Router PE1, configure BGP between the PE and CE routers as follows:

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface fe-1/0/0.0;
    route-distinguisher 10.255.14.175:3;
    vrf-import vpn-a-import;
    vrf-export vpn-a-export;
    protocols {
      bgp {
        group vpn-a-site1 {
          family inet {
            unicast {
              rib-group vpn-a-vpnab;
            }
          }
        }
        peer-as 1;
        neighbor 192.168.197.141;
      }
    }
  }
}
VPN-AB {
  instance-type vrf;
  interface fe-1/1/0.0;
  route-distinguisher 10.255.14.175:9;
  vrf-import vpn-ab-import;
  vrf-export vpn-ab-export;
  protocols {
    bgp {
      group vpn-ab-site1 {
        family inet {
          unicast {
            rib-group vpn-ab-vpn-a_and_vpn-b;
          }
        }
      }
      peer-as 9;
      neighbor 192.168.197.178;
    }
  }
}
```

```

VPN-B {
  instance-type vrf;
  interface fe-1/0/2.0;
  route-distinguisher 10.255.14.175:10;
  vrf-import vpnb-import;
  vrf-export vpnb-export;
  protocols {
    bgp {
      group vpnb-site1 {
        family inet {
          unicast {
            rib-group vpnb-vpnab;
          }
        }
      }
      neighbor 192.168.197.242 {
        peer-as 10;
      }
    }
  }
}

```

Configure OSPF between the PE and CE Routers

In this configuration example, routes learned from the OSPF session for VPN A are installed into the routing tables defined in the `vpna-vpnab` routing table group. For VPN AB, routes learned from the OSPF session are installed into the routing tables defined in the `vpnab-vpna_and_vpnb` routing table group. For VPN B, routes learned from the OSPF session are installed into the routing tables defined in the `vpnb-vpnab` routing table group.

The VRF import and export policies are similar to those defined in “Configure Static Routes between the PE and CE Routers” on page 185 and “Configure BGP between the PE and CE Routers” on page 190, except the export protocol is OSPF instead of BGP or a static route. Therefore, on all `vrf-export` policies, you use the `from protocol ospf` statement instead of the `from protocol <static | bgp>` statement.

On Router PE1, configure OSPF between the PE and CE routers as follows:

```

[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface fe-1/0/0.0;
    route-distinguisher 10.255.14.175:3;
    vrf-import vpna-import;
    vrf-export vpna-export;
    protocols {
      ospf {
        rib-group vpna-vpnab;
        export vpna-import;
        area 0.0.0.0 {
          interface fe-1/0/0.0;
        }
      }
    }
  }
}

```



```

VPN-AB {
  instance-type vrf;
  interface fe-1/1/0.0;
  route-distinguisher 10.255.14.175:9;
  vrf-import vpnab-import;
  vrf-export vpnab-export;
  protocols {
    ospf {
      rib-group vpnab-vpna_and_vpnb;
      export vpnab-import;
      area 0.0.0.0 {
        interface fe-1/1/0.0;
      }
    }
  }
}
VPN-B {
  instance-type vrf;
  interface fe-1/0/2.0;
  route-distinguisher 10.255.14.175:10;
  vrf-import vpnb-import;
  vrf-export vpnb-export;
  protocols {
    ospf {
      rib-group vpnb-vpnab;
      export vpnb-import;
      area 0.0.0.0 {
        interface fe-1/0/2.0;
      }
    }
  }
}
}

```

Configure Static, BGP, and OSPF Routes between the PE and CE Routers

This section shows how to configure the routes between the PE and CE routers using a combination of static routes, BGP, and OSPF, as follows:

- The connection between Router PE1 and Router CE1 uses static routing.
- The connection between Router PE1 and Router CE2 uses BGP.
- The connection between Router PE1 and Router CE3 uses OSPF.

Here, the configuration for VPN AB also includes a static route to CE1.

On Router PE1, configure a combination of static routing, BGP, and OSPF between the PE and CE routers:

```

[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface fe-1/0/0.0;
    route-distinguisher 10.255.14.175:3;
    vrf-import vpna-import;
    vrf-export vpna-export;
  }
}

```

```

routing-options {
  static {
    rib-group vpnab-vpnab;
    route 10.255.14.155/32 next-hop 192.168.197.141;
  }
}
}
VPN-AB {
  instance-type vrf;
  interface fe-1/1/0.0;
  route-distinguisher 10.255.14.175:9;
  vrf-import vpnab-import;
  vrf-export vpnab-export;
  protocols {
    bgp {
      group vpnab-site1 {
        family inet {
          unicast {
            rib-group vpnab-vpna_and_vpnab;
          }
        }
        export to-vpnab-site1;
        peer-as 9;
        neighbor 192.168.197.178;
      }
    }
  }
}
VPN-B {
  instance-type vrf;
  interface fe-1/0/2.0;
  route-distinguisher 10.255.14.175:10;
  vrf-import vpnb-import;
  vrf-export vpnb-export;
  protocols {
    ospf {
      rib-group vpnb-vpnab;
      export vpnb-import;
      area 0.0.0.1 {
        interface t3-0/3/3.0;
      }
    }
  }
}
}
policy-options {
  policy-statement to-vpnab-site1 {
    term a {
      from protocol static;
      then accept;
    }
    term b {
      from protocol bgp;
      then accept;
    }
    term c {
      then reject;
    }
  }
}

```

Configuring Overlapping VPNs Using Automatic Route Export

A problem with multiple routing instances is how to export routes between routing instances. This can be accomplished in JUNOS software by configuring routing table groups for each routing instance that needs to export routes to other routing tables. For information on how to configure overlapping VPNs using routing table groups, see “Configure Overlapping VPNs Using Routing Table Groups” on page 183.

However, using routing table groups has limitations:

- Routing table group configuration is complex. A unique routing table group must be defined for each routing instance that will export routes.
- You must also configure a unique routing table group for each protocol that will export routes.

To limit and sometimes eliminate the need to configure routing table groups in multiple routing instance topologies, you can use the functionality provided by the auto-export statement.

The auto-export statement is particularly useful for configuring overlapping VPNs—VPN configurations where more than one VRF lists the same community route target in its vrf-import policy. The auto-export statement finds out which routing tables to export routes from and import routes to by examining the existing policy configuration.

The auto-export statement automatically exports routes between the routing instances referencing a given route target community. When the auto-export statement is configured, a VRF target tree is constructed based on the vrf-import and vrf-export policies configured on the system. If a routing instance references a target in its vrf-import policy, it is added to the import list for the target. If it references a specific route target in its vrf-export policy, it is added to the export list for that target. Route targets where there is a single importer that matches a single exporter or with no importers or exporters are ignored.

Changes to routing tables that export route targets are tracked. When a route change occurs, the routing instance’s vpn-export policy is applied to the route. If it is allowed, the route is imported to all the import tables (subject to the vrf-import policy) of the route targets set by the export policy.

The sections that follow describe how to configure overlapping VPNs using the auto-export statement for interinstance export in addition to routing table groups:

- Configuring Overlapping VPNs with BGP and auto-export on page 196
- Configuring Overlapping VPNs and Additional Tables on page 197
- Configuring auto-export for all VRF Instances on page 198

Configuring Overlapping VPNs with BGP and auto-export

The following example provides the configuration for an overlapping VPN where BGP is used between the PE and CE routers.

Configure routing instance VPN-A as follows:

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface fe-1/0/0.0;
    route-distinguisher 10.255.14.175:3;
    vrf-import vpn-a-import;
    vrf-export vpn-a-export;
    routing-options {
      auto-export;
    }
    protocols {
      bgp {
        group vpn-a-site1 {
          peer-as 1;
          neighbor 192.168.197.141;
        }
      }
    }
  }
}
```

Configure routing instance VPN-AB as follows:

```
[edit]
routing-instances {
  VPN-AB {
    instance-type vrf;
    interface fe-1/1/0.0;
    route-distinguisher 10.255.14.175:9;
    vrf-import vpnab-import;
    vrf-export vpnab-export;
    routing-options {
      auto-export;
    }
    protocols {
      bgp {
        group vpnab-site1 {
          peer-as 9;
          neighbor 192.168.197.178;
        }
      }
    }
  }
}
```

For this configuration, the auto-export statement replaces the functionality that was provided by a routing table group configuration. However, sometimes additional configuration is required.

Since the vrf-import policy and the vrf-export policy from which the auto-export statement deduces the import and export matrix are configured on a per-instance basis, it is necessary to be able to enable or disable them for unicast and multicast, in case multicast network layer reachability information (NLRI) is configured.

Configuring Overlapping VPNs and Additional Tables

It might be necessary to use the auto-export statement between overlapping VPNs, but require that a subset of the routes learned from a VRF table be installed into the inet.0 table or in routing-instance.inet.2.

To support this type of scenario, where not all of the information needed is present in the vrf-import and vrf-export policies, you configure an additional list of routing tables using an additional routing table group.

To add routes from VPN-A and VPN-AB to inet.0 in the example described in “Configuring Overlapping VPNs with BGP and auto-export” on page 196, you need to include the following additional configuration statements:

Configure the routing options as follows:

```
[edit]
routing-options {
  rib-groups {
    inet-access {
      import-rib inet.0;
    }
  }
}
```

Configure routing instance VPN-A as follows:

```
[edit]
routing-instances {
  VPN-A {
    routing-options {
      auto-export {
        family inet {
          unicast {
            rib-group inet-access;
          }
        }
      }
    }
  }
}
```

Configure routing instance VPN-AB as follows:

```
[edit]
routing-instances {
  VPN-AB {
    routing-options {
      auto-export {
        family inet {
          unicast {
            rib-group inet-access;
          }
        }
      }
    }
  }
}
```

Routing table groups are used in this configuration differently from how they are generally used in JUNOS software. Routing table groups normally require that the exporting routing table be referenced as the primary import routing table in the routing table group. For this configuration, the restriction does not apply. The routing table group functions as an additional list of tables to which to export routes.

Configuring auto-export for all VRF Instances

The following configuration allows you to configure the auto-export statement for all of the routing instances in a configuration group:

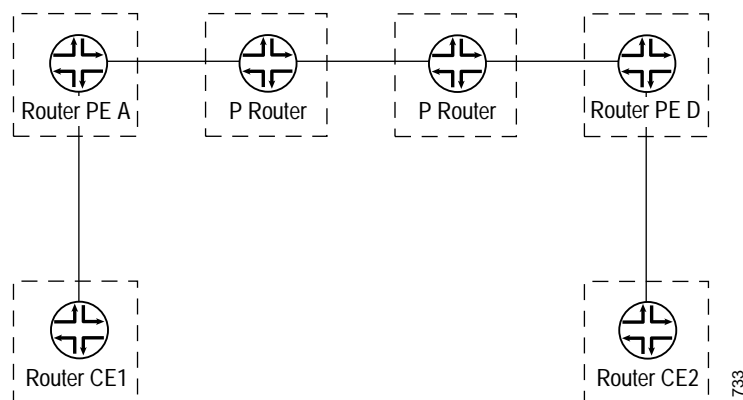
```
[edit]
groups {
  vrf-export-on {
    routing-instances {
      <*> {
        routing-options {
          auto-export;
        }
      }
    }
  }
}

apply-groups vrf-export-on;
```

Configure a GRE Tunnel Interface between PE Routers

This example shows how to configure a generic routing encapsulation (GRE) tunnel interface between PE routers to provide VPN connectivity. You can use this configuration to tunnel VPN traffic across a non-MPLS core network. The network topology used in this example is shown in Figure 24. Note that the provider (P) routers shown in this illustration do not run MPLS.

Figure 24: PE Router A and PE Router D Connected by a GRE Tunnel Interface



Configure the Routing Instance on Router A

Configure a routing instance on Router A as follows:

```
[edit routing-instances]
gre-config {
  instance-type vrf;
  interface fe-1/0/0.0;
  route-distinguisher 10.255.14.176:69;
  vrf-import import-config;
  vrf-export export-config;
  protocols {
    ospf {
      export import-config;
      area 0.0.0.0 {
        interface all;
      }
    }
  }
}
```

Configure the Routing Instance on Router D

Configure a routing instance on Router D as follows:

```
[edit routing-instances]
gre-config {
  instance-type vrf;
  interface fe-1/0/1.0;
  route-distinguisher 10.255.14.178:69;
  vrf-import import-config;
  vrf-export export-config;
  protocols {
    ospf {
      export import-config;
      area 0.0.0.0 {
        interface all;
      }
    }
  }
}
```

Configure MPLS, BGP, and OSPF on Router A

Although MPLS does not need to be configured on the P routers in this example, it is needed on the PE routers for the interface between the PE and CE routers and on the GRE interface (gr-1/1/0.0) linking the PE routers (Router A and Router D). Configure MPLS, BGP, and OSPF on Router A as follows:

```
[edit protocols]
mpls {
  interface all;
}
bgp {
  group pe-to-pe {
    type internal;
    neighbor 10.255.14.178 {
      family inet-vpn {
        unicast;
      }
    }
  }
}
ospf {
  area 0.0.0.0 {
    interface all;
    interface gr-1/1/0.0 {
      disable;
    }
  }
}
```

Configure MPLS, BGP, and OSPF on Router D

Although MPLS does not need to be configured on the P routers in this example, it is needed on the PE routers for the interface between the PE and CE routers and on the GRE interface (gr-1/1/0.0) linking the PE routers (Router D and Router A). Configure MPLS, BGP, and OSPF on Router D as follows:

```
[edit protocols]
mpls {
  interface all;
}
bgp {
  group pe-to-pe {
    type internal;
    neighbor 10.255.14.176 {
      family inet-vpn {
        unicast;
      }
    }
  }
}
```



```
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface all;
    interface fxp0.0 {
      disable;
    }
    interface gr-1/1/0.0 {
      disable;
    }
  }
}
```

Configure the Tunnel Interface on Router A

Configure the tunnel interface on Router A as follows (note that the tunnel is unnumbered):

```
[edit interfaces interface-name]
unit 0 {
  tunnel {
    source 10.255.14.176;
    destination 10.255.14.178;
  }
  family inet;
  family mpls;
}
```

Configure the Tunnel Interface on Router D

Configure the tunnel interface on Router D as follows (note that the tunnel is unnumbered):

```
[edit interfaces interface-name]
unit 0 {
  tunnel {
    source 10.255.14.178;
    destination 10.255.14.176;
  }
  family inet;
  family mpls;
}
```

Configure the Routing Options on Router A

As part of the routing options configuration for Router A, you need to configure routing table groups to enable VPN route resolution in the inet.3 routing table.

Configure the routing options on Router A as follows:

```
[edit routing-options]
interface-routes {
  rib-group inet if-rib;
}
rib inet.3 {
  static {
    route 10.255.14.178/32 next-hop gr-1/1/0.0;
  }
}
```

```

    rib-groups {
      if-rib {
        import-rib [ inet.0 inet.3 ];
      }
    }

```

Configure the Routing Options on Router D

As part of the routing options configuration for Router D, you need to configure routing table groups to enable VPN route resolution in the inet.3 routing table.

Configure the routing options on Router D as follows:

```

[edit routing-options]
interface-routes {
  rib-group inet if-rib;
}
rib inet.3 {
  static {
    route 10.255.14.176/32 next-hop gr-1/1/0.0;
  }
}
rib-groups {
  if-rib {
    import-rib [ inet.0 inet.3 ];
  }
}

```

Configuration Summary for Router A

Configure the Routing Instance	<pre> gre-config { instance-type vrf; interface fe-1/0/0.0; route-distinguisher 10.255.14.176:69; vrf-import import-config; vrf-export export-config; protocols { ospf { export import-config; area 0.0.0.0 { interface all; } } } } </pre>
---------------------------------------	---

Configure MPLS	<pre> mpls { interface all; } </pre>
-----------------------	--

Configure BGP	<pre> bgp { traceoptions { file bgp.trace world-readable; flag update detail; } group pe-to-pe { type internal; neighbor 10.255.14.178 { family inet-vpn { unicast; } } } } </pre>
Configure OSPF	<pre> ospf { area 0.0.0.0 { interface all; interface gr-1/1/0.0 { disable; } } } </pre>
Configure the Tunnel Interface	<pre> interface-name { unit 0 { tunnel { source 10.255.14.176; destination 10.255.14.178; } family inet; } } </pre>
Configure Routing Options	<pre> interface-routes { rib-group inet if-rib; } rib inet.3 { static { route 10.255.14.178/32 next-hop gr-1/1/0.0; } } rib-groups { if-rib { import-rib [inet.0 inet.3]; } } </pre>

Configuration Summary for Router D

Configure the Routing Instance

```
gre-config {
  instance-type vrf;
  interface fe-1/0/1.0;
  route-distinguisher 10.255.14.178:69;
  vrf-import import-config;
  vrf-export export-config;
  protocols {
    ospf {
      export import-config;
      area 0.0.0.0 {
        interface all;
      }
    }
  }
}
```

Configure MPLS

```
mpls {
  interface all;
}
```

Configure BGP

```
bgp {
  group pe-to-pe {
    type internal;
    neighbor 10.255.14.176 {
      family inet-vpn {
        unicast;
      }
    }
  }
}
```

Configure OSPF

```
ospf {
  traffic-engineering;
  area 0.0.0.0 {
    interface all;
    interface fxp0.0 {
      disable;
    }
    interface gr-1/1/0.0 {
      disable;
    }
  }
}
```

Configure the Tunnel Interface

```
interface-name {
  unit 0 {
    tunnel {
      source 10.255.14.178;
      destination 10.255.14.176;
    }
    family inet;
  }
}
```

Configure the Routing Options

```

interface-routes {
  rib-group inet if-rib;
}
rib inet.3 {
  static {
    route 10.255.14.176/32 next-hop gr-1/1/0.0;
  }
}
rib-groups {
  if-rib {
    import-rib [ inet.0 inet.3 ];
  }
}

```

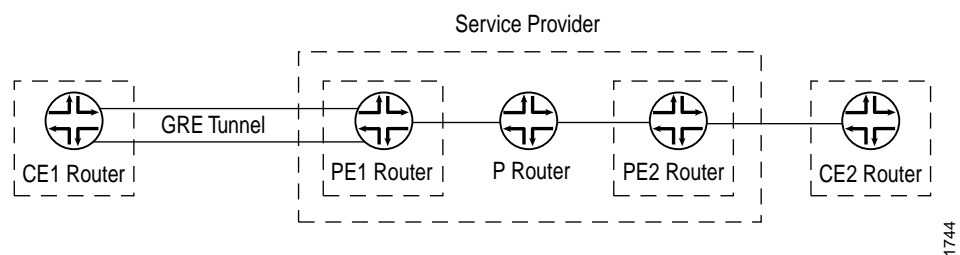
Configure a GRE Tunnel Interface between a PE and CE Router

This example shows how to configure a GRE tunnel interface between a PE router and a CE router. You can use this configuration to tunnel VPN traffic across a non-MPLS core network. The network topology used in this example is shown in Figure 25.

For this example, complete the procedures described in the following sections:

- Configure the Routing Instance without the Encapsulating Interface on page 205
- Configure the Routing Instance with the Encapsulating Interface on page 207
- Configure the GRE Tunnel Interface on Router CE1 on page 208

Figure 25: GRE Tunnel between the CE Router and the PE Router



Configure the Routing Instance without the Encapsulating Interface

You can configure the routing instance either with or without the encapsulating interface. The sections that follow describe how to configure the routing instance without it:

- Configure the Routing Instance on Router PE1 on page 206
- Configure the GRE Tunnel Interface on Router PE1 on page 206
- Configure the Encapsulation Interface on Router PE1 on page 206

Configure the Routing Instance on Router PE1

Configure the routing instance on Router PE1 as follows:

```
[edit routing-instances]
vpna {
  instance-type vrf;
  interface gr-1/2/0.0;
  route-distinguisher 10.255.14.174:1;
  vrf-import vpna-import;
  vrf-export vpna-export;
  protocols {
    bgp {
      group vpna {
        type external;
        peer-as 100;
        as-override;
        neighbor 10.49.2.1;
      }
    }
  }
}
```

Configure the GRE Tunnel Interface on Router PE1

Configure the GRE tunnel interface on Router PE1 as follows:

```
[edit interfaces gr-1/2/0]
unit 0 {
  tunnel {
    source 192.168.197.249;
    destination 192.168.197.250;
  }
  family inet {
    address 10.49.2.2/30;
  }
}
```

In this example, interface t3-0/1/3 acts as the encapsulating interface for the GRE tunnel.

Configure the Encapsulation Interface on Router PE1

Configure the encapsulation interface on Router PE1 as follows:

```
[edit interfaces t3-0/1/3]
unit 0 {
  family inet {
    address 192.168.197.249/30;
  }
}
```

Configure the Routing Instance with the Encapsulating Interface

If the tunnel encapsulating interface, t3-0/1/3, is also configured under the routing instance, then you need to specify the name of that routing instance under the interface definition. The system uses this routing instance to search for the tunnel destination address.

To configure the routing instance with the encapsulating interface, complete the procedures described in the following sections:

- Configure the Routing Instance on Router PE1 on page 207
- Configure the GRE Tunnel Interface on Router PE1 on page 207
- Configure the Encapsulation Interface on Router PE1 on page 208

Configure the Routing Instance on Router PE1

If you configure the tunnel encapsulating interface under the routing instance, then configure the routing instance on Router PE1 as follows:

```
[edit routing-instances]
vpna {
  instance-type vrf;
  interface gr-1/2/0.0;
  interface t3-0/1/3.0;
  route-distinguisher 10.255.14.174:1;
  vrf-import vpna-import;
  vrf-export vpna-export;
  protocols {
    bgp {
      group vpna {
        type external;
        peer-as 100;
        as-override;
        neighbor 10.49.2.1;
      }
    }
  }
}
```

Configure the GRE Tunnel Interface on Router PE1

Configure the GRE tunnel interface on Router PE1 as follows:

```
[edit interfaces gr-1/2/0]
unit 0 {
  tunnel {
    source 192.168.197.249;
    destination 192.168.197.250;
    routing-instance {
      destination vpna;
    }
  }
  family inet {
    address 10.49.2.2/30;
  }
}
```

Configure the Encapsulation Interface on Router PE1

Configure the encapsulation interface on Router PE1 as follows:

```
[edit interfaces t3-0/1/3]
unit 0 {
  family inet {
    address 192.168.197.249/30;
  }
}
```

Configure the GRE Tunnel Interface on Router CE1

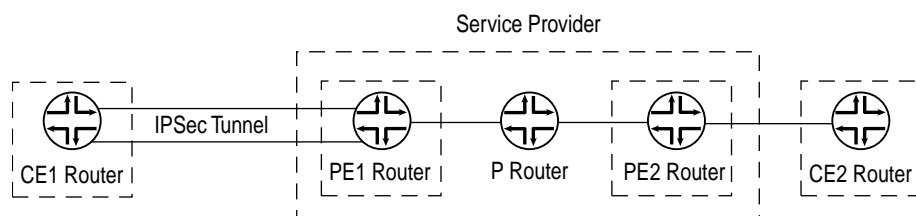
Configure the GRE tunnel interface on Router CE1 as follows:

```
[edit interfaces gr-1/2/0]
unit 0 {
  tunnel {
    source 192.168.197.250;
    destination 192.168.197.249;
  }
  family inet {
    address 10.49.2.1/30;
  }
}
```

Configure an ES Tunnel Interface between a PE and CE Router

This example shows how to configure an ES tunnel interface between a PE router and a CE router in a Layer 3 VPN. The network topology used in this example is shown in Figure 26.

Figure 26: ES Tunnel Interface (IPSec Tunnel) between the CE Router and the PE Router



1745

To configure this example, complete the steps in the following sections:

- Configure IPSec on Router PE1 on page 209
- Configure the Routing Instance without the Encapsulating Interface on page 209
- Configure the Routing Instance with the Encapsulating Interface on page 210
- Configure the ES Tunnel Interface on Router CE1 on page 212
- Configure IPSec on Router CE1 on page 212

Configure IPsec on Router PE1

Configure IP Security (IPsec) on Router PE1 as follows:

```
[edit security]
ipsec {
  security-association sa-esp-manual {
    mode tunnel;
    manual {
      direction bidirectional {
        protocol esp;
        spi 16000;
        authentication {
          algorithm hmac-md5-96;
          key ascii-text "$9$ABULt1heK87dsWLDk.P3nrevM7V24ZHkPaZ/tpOcSvWLNwgZUH";
        }
        encryption {
          algorithm des-cbc;
          key ascii-text "$9$/H8Q90lyrvL7VKMZjHqQzcyleLN";
        }
      }
    }
  }
}
```

Configure the Routing Instance without the Encapsulating Interface

You can configure the routing instance on Router PE1 with or without the encapsulating interface (t3-0/1/3 in this example). The following sections describe how to configure the routing instance without it:

- Configure the Routing Instance on Router PE1 on page 210
- Configure the ES Tunnel Interface on Router PE1 on page 210
- Configure the Encapsulating Interface for the ES Tunnel on Router PE1 on page 210

Configure the Routing Instance on Router PE1

Configure the routing instance on Router PE1 as follows:

```
[edit routing-instances]
vpna {
  instance-type vrf;
  interface es-1/2/0.0;
  route-distinguisher 10.255.14.174:1;
  vrf-import vpna-import;
  vrf-export vpna-export;
  protocols {
    bgp {
      group vpna {
        type external;
        peer-as 100;
        as-override;
        neighbor 10.49.2.1;
      }
    }
  }
}
```

Configure the ES Tunnel Interface on Router PE1

Configure the ES tunnel interface on Router PE1 as follows:

```
[edit interfaces es-1/2/0]
unit 0 {
  tunnel {
    source 192.168.197.249;
    destination 192.168.197.250;
  }
  family inet {
    address 10.49.2.2/30;
    ipsec-sa sa-esp-manual;
  }
}
```

Configure the Encapsulating Interface for the ES Tunnel on Router PE1

For this example, interface t3-0/1/3 is the encapsulating interface for the ES tunnel. Configure interface t3-0/1/3 as follows:

```
[edit interfaces t3-0/1/3]
unit 0 {
  family inet {
    address 192.168.197.249/30;
  }
}
```

Configure the Routing Instance with the Encapsulating Interface

If the tunnel encapsulating interface, t3-0/1/3, is also configured under the routing instance, you need to specify the routing instance name under the interface definition. The system uses this routing instance to search for the tunnel destination address for the IPSec tunnel using manual security association.

The following sections describe how to configure the routing instance with the encapsulating interface:

- Configure the Routing Instance on Router PE1 on page 211
- Configure the ES Tunnel Interface on Router PE1 on page 211
- Configure the Encapsulating Interface on Router PE1 on page 212

Configure the Routing Instance on Router PE1

Configure the routing instance on Router PE1 (including the tunnel encapsulating interface) as follows:

```
[edit routing-instances]
vpna {
  instance-type vrf;
  interface es-1/2/0.0;
  interface t3-0/1/3.0;
  route-distinguisher 10.255.14.174:1;
  vrf-import vpna-import;
  vrf-export vpna-export;
  protocols {
    bgp {
      group vpna {
        type external;
        peer-as 100;
        as-override;
        neighbor 10.49.2.1;
      }
    }
  }
}
```

Configure the ES Tunnel Interface on Router PE1

Configure the ES tunnel interface on Router PE1 as follows:

```
[edit interfaces es-1/2/0]
unit 0 {
  tunnel {
    source 192.168.197.249;
    destination 192.168.197.250;
    routing-instance {
      destination vpna;
    }
  }
  family inet {
    address 10.49.2.2/30;
    ipsec-sa sa-esp-manual;
  }
}
```

Configure the Encapsulating Interface on Router PE1

Configure the encapsulating interface on Router PE1 as follows:

```
[edit interfaces t3-0/1/3]
unit 0 {
  family inet {
    address 192.168.197.249/30;
  }
}
```

Configure the ES Tunnel Interface on Router CE1

Configure the ES tunnel interface on Router CE1 as follows:

```
[edit interfaces es-1/2/0]
unit 0 {
  tunnel {
    source 192.168.197.250;
    destination 192.168.197.249;
  }
  family inet {
    address 10.49.2.1/30;
    ipsec-sa sa-esp-manual;
  }
}
```

Configure IPsec on Router CE1

Configure IPsec on Router CE1 as follows:

```
[edit security]
ipsec {
  security-association sa-esp-manual {
    mode tunnel;
    manual {
      direction bidirectional {
        protocol esp;
        spi 16000;
        authentication {
          algorithm hmac-md5-96;
          key ascii-text "$9$ABULt1heK87dsWLDk.P3nrevM7V24ZHkPaZ/tpOcSvWLNwgZUH";
        }
        encryption {
          algorithm des-cbc;
          key ascii-text "$9$/H8Q90IyrvL7VKMZjHqQzcyleLN";
        }
      }
    }
  }
}
```

Configure SCU and DCU for Layer 3 VPNs

For information on how to configure source class usage (SCU) for a Layer 3 VPN loopback interface, see the *JUNOS Internet Software Configuration Guide: Network Management*.

For information on how to configure SCU and destination class usage (DCU) to count packets on Layer 3 VPNs, see the *JUNOS Internet Software Configuration Guide: Interfaces and Class of Service*.

.....

Chapter 10

Layer 3 VPN Internet Access Examples

JUNOS software supports Internet access from a Layer 3 virtual private network (VPN). This chapter provides examples that demonstrate how to configure a provider edge (PE) router to provide Internet access to customer edge (CE) routers in a VPN. The method you use depends on the needs and specifications of the individual network. To provide Internet access through a Layer 3 VPN, you need to configure policies on the PE router. You also need to configure the next-table keyword at the [edit routing-instances *routing-instance-name* routing-options static route] hierarchy level. When configured, this statement can point a default route from the VPN table (routing instance) to the main routing table (default instance) inet.0. The main routing table stores all Internet routes and is where final route resolution occurs.

There are several ways to configure a PE router to provide CE routers access to the Internet. These types of access are described in the following sections:

- Non-VRF Internet Access on page 215—Internet and VPN access are separate. The CE routers access the Internet independently of the PE routers.
- Distributed Internet Access on page 216—The PE router provides Internet access to the CE routers. Internet route information is stored in the PE router's main routing table.
- Centralized Internet Access on page 241—Some of the CE routers are specially configured to provide Internet access to the other CE routers within the VPN.

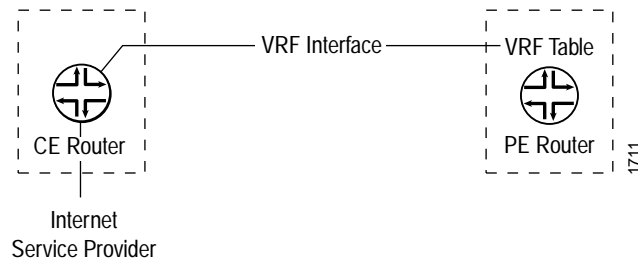
Non-VRF Internet Access

The following sections describe ways to provide Internet access to a CE router in a Layer 3 VPN without using the VPN routing and forwarding (VRF) interface. Because these methods effectively bypass the Layer 3 VPN, they are not discussed in detail.

CE Router Accesses Internet Independently of the PE Router

In this configuration, the PE router does not provide the Internet access. The CE router sends Internet traffic either to another service provider, or to the same service provider but a different router. The PE router handles Layer 3 VPN traffic only (see Figure 27).

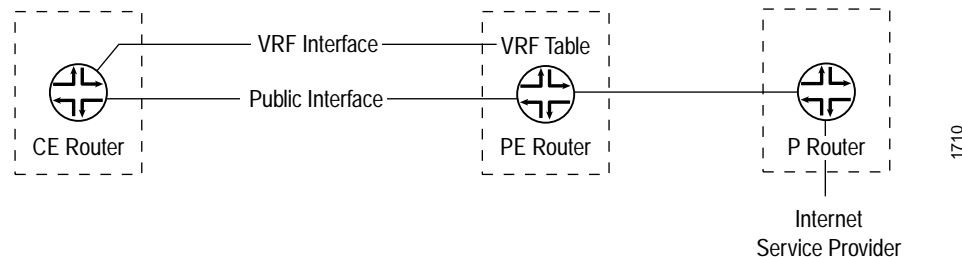
Figure 27: PE Router Does Not Provide Internet Access



PE Router Provides Layer 2 Internet Service

In this configuration, the PE router acts as a Layer 2 device, providing a Layer 2 connection (such as circuit cross-connect [CCC]) to another router that has a full set of Internet routes. The CE router can use just one physical interface and two logical interfaces to the PE router, or it can use multiple physical interfaces to the PE router (see Figure 28).

Figure 28: PE Router Connects to a Router Connected to the Internet



Distributed Internet Access

In this scenario, the PE routers provide Internet access to the CE routers. In the examples that follow, it is assumed that the Internet routes (or defaults) are present in the inet.0 table of the PE routers that provide Internet access to selected CE routers.

When accessing the Internet from a VPN, Network Address Translation (NAT) must be performed between the VPN's private addresses and the public addresses used on the Internet unless the VPN is using the public address space. This section includes several examples of how to provide Internet access for VPNs, most of which require that the CE routers perform the address translation. The "Route Internet Traffic through a Separate NAT Device" example, however, requires that the service provider supply the NAT functionality using a NAT device connected to the PE router.

This section includes the following examples:

- Route VPN and Internet Traffic through Different Interfaces on page 217
- Route VPN and Outgoing Internet Traffic through the Same Interface and Route Return Internet Traffic through a Different Interface on page 224

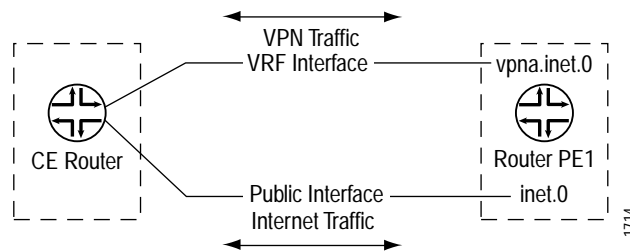
- Route VPN and Internet Traffic through the Same Interface Bidirectionally (VPN Has Public Addresses) on page 225
- Route VPN and Internet Traffic through the Same Interface Bidirectionally (VPN Has Private Addresses) on page 229
- Route Internet Traffic through a Separate NAT Device on page 233

In all of the examples, the VPN's public IP address pool (whose entries correspond to the translated private addresses) must be added to the inet.0 table and propagated to the Internet routers to receive reverse traffic from public destinations.

Route VPN and Internet Traffic through Different Interfaces

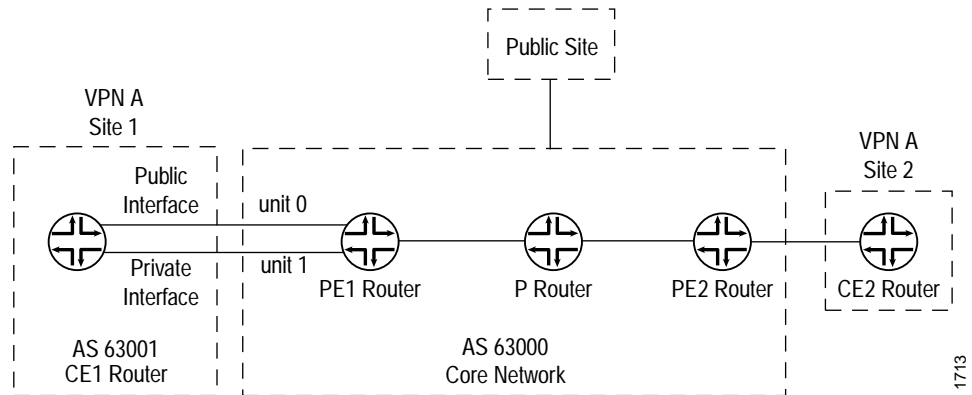
In this example, VPN and Internet traffic are routed through different interfaces. The CE router sends the VPN traffic through the VPN interface and sends the Internet traffic through a separate interface that is part of the main routing table on Router PE1 (the CE router can use either one physical interface with two logical units or two physical interfaces). NAT also occurs on the CE router (see Figure 29).

Figure 29: Routing VPN and Internet Traffic through Different Interfaces



The PE router is configured to install and advertise the public IP address pool for the VPN to other core routers (for return traffic). The VPN traffic is routed normally. Figure 30 illustrates the PE router's VPN configuration.

Figure 30: Example of Internet Traffic Routed through Separate Interfaces



The configuration in this example has the following features:

- Router PE1 uses two logical interfaces to connect to Router CE1 using Frame Relay encapsulation.
- The routing protocol between Router PE1 and Router CE1 is the external Border Gateway Protocol (EBGP).
- Router CE1's public IP address pool is 10.12.1.1-10.12.1.254 (10.12.1.0/24).
- The next-hop-self setting is derived from the fix-nh policy statement on Router PE1. PE routers are forced to use next-hop-self so that next-hop resolution is done only for the PE router's loopback address for non-VPN routes (by default, VPN-Internet Protocol version 4 [IPv4] routes are sent using next-hop-self).

You can configure Router CE1 with a static default route pointing to its public interface for everything else.

Configure Interfaces on Router PE1

Configure an interface to handle VPN traffic and an interface to handle Internet traffic as follows:

```
[edit]
interfaces {
  t3-0/2/0 {
    dce;
    encapsulation frame-relay;
    unit 0 {
      description "to CE1 VPN interface";
      dlci 10;
      family inet {
        address 192.168.197.13/30;
      }
    }
    unit 1 {
      description "to CE1 public interface";
      dlci 20;
      family inet {
        address 192.168.198.201/30;
      }
    }
  }
}
```

Configure Routing Options on Router PE1

Configure a static route on Router PE1 to install a route to the CE router's public IP address pool in inet.0 as follows:

```
[edit]
routing-options {
  static {
    route 10.12.1.0/24 next-hop 192.168.198.202;
  }
}
```

Configure BGP, IS-IS, and LDP Protocols on Router PE1

Configure BGP on Router PE1 to allow non-VPN and VPN peering and to advertise the VPN's public IP address pool as follows:

```
[edit]
protocols {
  bgp {
    group pe-pe {
      type internal;
      local-address 10.255.14.171;
      family inet {
        any;
      }
      family inet-vpn {
        any;
      }
    }
  }
}
```

```

    export [ fix-nh redistrib-static];
    neighbor 10.255.14.177;
    neighbor 10.255.14.179;
  }
}

```

Configure Intermediate System-to-Intermediate System (IS-IS) on Router PE1 to allow access to internal routes as follows:

```

[edit protocols]
isis {
  level 1 disable;
  interface so-0/0/0.0;
  interface lo0.0;
}

```

Configure Label Distribution Protocol (LDP) on Router PE1 to tunnel VPN routes as follows:

```

[edit protocols]
ldp {
  interface so-0/0/0.0;
}

```

Configure a Routing Instance on Router PE1

Configure a routing instance on Router PE1 as follows:

```

[edit]
routing-instances {
  vpn {
    instance-type vrf;
    interface t3-0/2/0.0;
    route-distinguisher 10.255.14.171:100;
    vrf-import vpn-import;
    vrf-export vpn-export;
    protocols {
      bgp {
        group to-CE1 {
          peer-as 63001;
          neighbor 192.168.197.14;
        }
      }
    }
  }
}

```

Configure Policy Options on Router PE1

You need to configure policy options on Router PE1. The fix-nh policy statement sets next-hop-self for all non-VPN routes:

```
[edit]
policy-options {
  policy-statement fix-nh {
    then {
      next-hop self;
    }
  }
}
```

The redist-static policy statement advertises the VPN's public IP address pool:

```
[edit policy-options]
policy-statement redist-static {
  term a {
    from {
      protocol static;
      route-filter 10.12.1.0/24 exact;
    }
    then accept;
  }
  term b {
    then reject;
  }
}
```

Configure import and export policies for vpna as follows:

```
[edit policy-options]
policy-statement vpna-import {
  term a {
    from {
      protocol bgp;
      community vpna-comm;
    }
    then accept;
  }
  term b {
    then reject;
  }
}
policy-statement vpna-export {
  term a {
    from protocol bgp;
    then {
      community add vpna-comm;
      accept;
    }
  }
  term b {
    then reject;
  }
}
community vpna-comm members target:63000:100;
}
```

Traffic Routed by Different Interfaces Configuration Summarized by Router

Router PE1

```

Interfaces    interfaces {
                  t3-0/2/0 {
                    dce;
                    encapsulation frame-relay;
                    unit 0 {
                      description "to CE1 VPN interface";
                      dlci 10;
                      family inet {
                        address 192.168.197.13/30;
                      }
                    }
                    unit 1 {
                      description "to CE1 public interface";
                      dlci 20;
                      family inet {
                        address 192.168.198.201/30;
                      }
                    }
                  }
                }

Routing Options routing-options {
                  static {
                    route 10.12.1.0/24 next-hop 192.168.198.202;
                  }
                }

BGP Protocol  protocols {
                  bgp {
                    group pe-pe {
                      type internal;
                      local-address 10.255.14.171;
                      family inet {
                        any;
                      }
                      family inet-vpn {
                        any;
                      }
                      export [ fix-nh redist-static];
                      neighbor 10.255.14.177;
                      neighbor 10.255.14.179;
                    }
                  }
                }

IS-IS Protocol isis {
                  level 1 disable;
                  interface so-0/0/0.0;
                  interface lo0.0;
                }

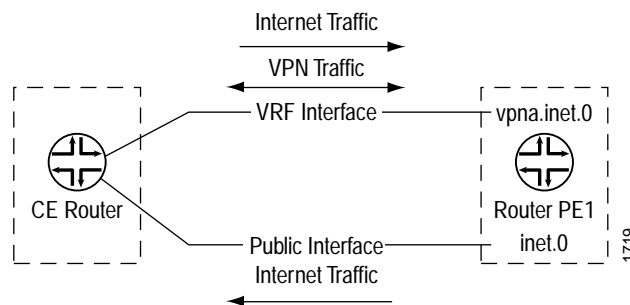
LDP Protocol  ldp {
                  interface so-0/0/0.0;
                }
  
```

Routing Instance	<pre> routing-instances { vpn { instance-type vrf; interface t3-0/2/0.0; route-distinguisher 10.255.14.171:100; vrf-import vpn-import; vrf-export vpn-export; protocols { bgp { group to-CE1 { peer-as 63001; neighbor 192.168.197.14; } } } } } </pre>
Policy Options/Policy Statements	<pre> policy-options { policy-statement fix-nh { then { next-hop self; } } policy-statement redist-static { term a { from { protocol static; route-filter 10.12.1.0/24 exact; } then accept; } term b { then reject; } } } </pre>
Import and Export Policies	<pre> policy-statement vpn-import { term a { from { protocol bgp; community vpn-comm; } then accept; } term b { then reject; } } policy-statement vpn-export { term a { from protocol bgp; then { community add vpn-comm; accept; } } term b { then reject; } } community vpn-comm members target:63000:100; } </pre>

Route VPN and Outgoing Internet Traffic through the Same Interface and Route Return Internet Traffic through a Different Interface

In this example, the CE sends VPN and Internet traffic through the same interface but receives return Internet traffic through a different interface. The PE router has a default route in the VRF table pointing to the main routing table inet.0. It routes the VPN public IP address pool (return Internet traffic) through a different interface in inet.0 (see Figure 31). The CE router still performs NAT functions.

Figure 31: VPN and Outgoing Internet Traffic Routed through the Same Interface and Return Internet Traffic Routed through a Different Interface



Configuration for Router PE1

This example has the same configuration as Router PE1 in “Route VPN and Internet Traffic through Different Interfaces” on page 217. It uses the topology shown in Figure 30, “Example of Internet Traffic Routed through Separate Interfaces” on page 218. The default route to the VPN routing table is configured differently. At the [edit routing-options] hierarchy level, you configure a default static route that is installed in vpna.inet.0 and points to inet.0 for resolution:

```

[edit routing-options]
routing-instances {
  vpna {
    instance-type vrf;
    interface t3-0/2/0.0;
    route-distinguisher 10.255.14.171:100;
    vrf-import vpna-import;
    vrf-export vpna-export;
    routing-options {
      static {
        route 0.0.0.0/0 next-table inet.0;
      }
    }
  }
  protocols {
    bgp {
      group to-CE1 {
        peer-as 63001;
        neighbor 192.168.197.14;
      }
    }
  }
}

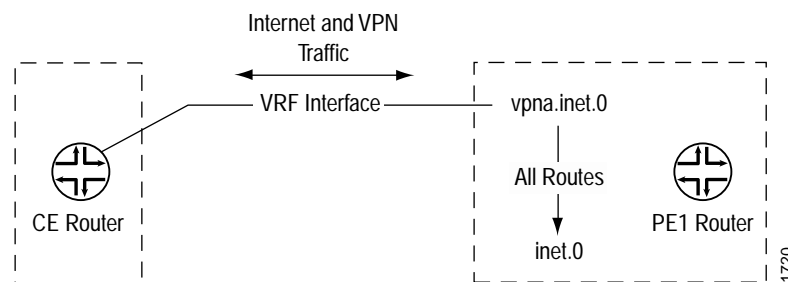
```


You also need to change the configuration of Router CE1 (from the configuration that works with the configuration for Router PE1 described in “Route VPN and Internet Traffic through Different Interfaces” on page 217) to account for the differences in the configuration of the PE routers.

Route VPN and Internet Traffic through the Same Interface Bidirectionally (VPN Has Public Addresses)

This section shows how to configure a single logical interface to handle VPN and Internet traffic traveling both to and from the Internet and the CE router. This interface can handle both VPN and Internet traffic as long as there are no private addresses in the VPN. The VPN routes received from the CE router are added to the main routing table inet.0 using routing table groups. This allows the PE router to attract the return traffic from the Internet (see Figure 32).

Figure 32: Interface Configured to Carry Both Internet and VPN Traffic



In this example, the CE router does not need to perform NAT because all the VPN routes are public. The CE router has a single interface to the PE router, to which it advertises VPN routes. The PE router has a default route in the VRF table pointing to the main routing table inet.0. The PE router also imports VPN routes received from the CE router into inet.0 using routing table groups.

The following configuration for Router PE1 uses the same topology as in “Route VPN and Internet Traffic through Different Interfaces” on page 217. This configuration uses a single logical interface (instead of two) between Router PE1 and Router CE1.

Configure Routing Options on Router PE1

Configure a routing table group definition for installing VPN routes in routing table groups vpna.inet.0 and inet.0 as follows:

```
[edit]
routing-options {
  rib-groups {
    vpna-to-inet0 {
      import-rib [ vpna.inet.0 inet.0 ];
    }
  }
}
```

Configure Routing Protocols on Router PE1

Configure the Multiprotocol Label Switching (MPLS), BGP, IS-IS, and LDP protocols on Router PE1. This configuration does not include the policy redistribute-static statement at the [edit protocols bgp group pe-pe] hierarchy level. The VPN routes are sent directly to IBGP.

Configure BGP on Router PE1 to allow non-VPN and VPN peering, and to advertise the VPN's public IP address pool as follows:

```
[edit]
protocols {
  mpls {
    interface t3-0/2/0.0;
  }
  bgp {
    group pe-pe {
      type internal;
      local-address 10.255.14.171;
      family inet {
        any;
      }
      family inet-vpn {
        any;
      }
      export fix-nh;
      neighbor 10.255.14.177;
      neighbor 10.255.14.173;
    }
  }
  isis {
    level 1 disable;
    interface so-0/0/0.0;
    interface lo0.0;
  }
  ldp {
    interface so-0/0/0.0;
  }
}
```

Configure the Routing Instance on Router PE1

This section describes how to configure the routing instance on Router PE1. The static route defined in the routing-options statement directs Internet traffic from the CE router to the inet.0 routing table. The routing table group defined by the rib-group vpn-to-inet0 statement adds the VPN routes to inet.0.

Configure the routing instance on Router PE1 as follows:

```
[edit]
routing-instances {
  vpn {
    instance-type vrf;
    interface t3-0/2/0.0;
    route-distinguisher 10.255.14.171:100;
    vrf-import vpn-import;
    vrf-export vpn-export;
    routing-options {
      static {
        route 0.0.0.0/0 next-table inet.0;
      }
    }
    protocols {
      bgp {
        group to-CE1 {
          family inet {
            unicast {
              rib-group vpn-to-inet0;
            }
          }
        }
        peer-as 63001;
        neighbor 192.168.197.14;
      }
    }
  }
}
```

Note that you must configure Router CE1 to forward all traffic to Router PE1 using a default route. Alternatively, the default route can be advertised from Router PE1 to Router CE1 with EBGp.

Traffic Routed through the Same Interface Bidirectionally Configuration Summarized by Router

Router PE1

This example uses the same configuration as in “Route VPN and Internet Traffic through Different Interfaces” on page 217. This configuration uses a single logical interface (instead of two) between Router PE1 and Router CE1.

```
Routing Options routing-options {
  rib-groups {
    vpn-to-inet0 {
      import-rib [ vpn.inet.0 inet.0 ];
    }
  }
}
```

```

Routing Protocols protocols {
    mpls {
        interface t3-0/2/0.0;
    }
    bgp {
        group pe-pe {
            type internal;
            local-address 10.255.14.171;
            family inet {
                any;
            }
            family inet-vpn {
                any;
            }
            export fix-nh;
            neighbor 10.255.14.177;
            neighbor 10.255.14.173;
        }
    }
    isis {
        level 1 disable;
        interface so-0/0/0.0;
        interface lo0.0;
    }
    ldp {
        interface so-0/0/0.0;
    }
}

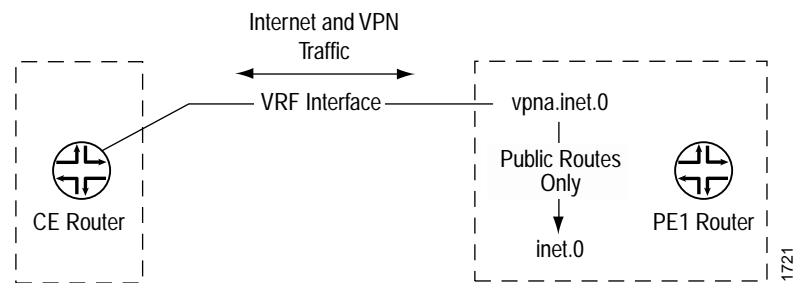
Routing Instance routing-instances {
    vpna {
        instance-type vrf;
        interface t3-0/2/0.0;
        route-distinguisher 10.255.14.171:100;
        vrf-import vpna-import;
        vrf-export vpna-export;
        routing-options {
            static {
                route 0.0.0.0/0 next-table inet.0;
            }
        }
        protocols {
            bgp {
                group to-CE1 {
                    family inet {
                        unicast {
                            rib-group vpna-to-inet0;
                        }
                    }
                }
                peer-as 63001;
                neighbor 192.168.197.14;
            }
        }
    }
}

```

Route VPN and Internet Traffic through the Same Interface Bidirectionally (VPN Has Private Addresses)

The example in this section shows how to route VPN and Internet traffic through the same interface in both directions (from the CE router to the Internet and from the Internet to the CE router). The VPN in this example has private addresses. If you can configure EBGp on the CE router, you can configure a PE router using the configuration outlined in “Route VPN and Internet Traffic through the Same Interface Bidirectionally (VPN Has Public Addresses)” on page 225, even if the VPN has private addresses. In the example described in this section, the CE router uses separate communities to advertise its VPN routes and public routes. The PE router selectively imports only the public routes into the inet.0 routing table. This configuration ensures that return traffic from the Internet uses the same interface between the PE and CE routers as that used by VPN traffic going out to public Internet addresses (see Figure 33).

Figure 33: VPN and Internet Traffic Routed through the Same Interface



In this example, the CE router has one interface and a BGP session with the PE router, and it tags VPN routes and Internet routes with different communities. The PE router has one interface, selectively imports routes for the VPN’s public IP address pool into inet.0, and has a default route in the VRF routing table pointing to inet.0.

Configure Routing Options for Router PE1

On Router PE1, you need to configure a routing table group to install VPN routes in the vpna.inet.0 and inet.0 routing tables:

```
[edit]
routing-options {
  rib-groups {
    vpna-to-inet0 {
      import-rib [ vpna.inet.0 inet.0 ];
    }
  }
}
```

Configure a Routing Instance for Router PE1

On Router PE1, you need to configure a routing instance. As part of the configuration for the routing instance, you need to configure a static route that is installed in `vpn1.inet.0` and is pointed at `inet.0` for resolution.

```
[edit]
routing-instances {
  vpn1 {
    instance-type vrf;
    interface t3-0/2/0.0;
    route-distinguisher 10.255.14.171:100;
    vrf-import vpn1-import;
    vrf-export vpn1-export;
    routing-options {
      static {
        route 0.0.0.0/0 next-table inet.0;
      }
    }
  }
}
```

At the `[edit routing-instances protocols bgp]` hierarchy level, configure a policy (`import-public-addr-to-inet0`) to import public routes into `inet.0` and a routing table group (`vpn1-to-inet0`) to allow BGP to install routes into multiple routing tables (`vpn1.inet.0` and `inet.0`):

```
[edit routing-instances]
protocols {
  bgp {
    group to-CE1 {
      import import-public-addr-to-inet0;
      family inet {
        unicast {
          rib-group vpn1-to-inet0;
        }
      }
    }
    peer-as 63001;
    neighbor 192.168.197.14;
  }
}
}
```

Configure Policy Options for Router PE1

Configure the policy options for Router PE1 to accept all routes initially (term a) and then to install routes with a public-comm community into routing table inet.0 (term b):

```
[edit]
policy-options {
  policy-statement import-public-addr-to-inet0 {
    term a {
      from {
        protocol bgp;
        rib vpn.a.inet.0;
        community [ public-comm private-comm ];
      }
      then accept;
    }
    term b {
      from {
        protocol bgp;
        community public-comm;
      }
      to rib inet.0;
      then accept;
    }
    term c {
      then reject;
    }
  }
  community private-comm members target:1:333;
  community public-comm members target:1:111;
  community vpn.a-comm members target:63000:100;
}
```

Traffic Routed by the Same Interface Bidirectionally (VPN Has Private Addresses) Configuration Summarized by Router

Router PE1

```
Routing Options  routing-options {
                  rib-groups {
                    vpn.a-to-inet0 {
                      import-rib [ vpn.a.inet.0 inet.0 ];
                    }
                  }
                }

Routing Instances routing-instances {
                  vpn.a {
                    instance-type vrf;
                    interface t3-0/2/0.0;
                    route-distinguisher 10.255.14.171:100;
                    vrf-import vpn.a-import;
                    vrf-export vpn.a-export;
                    routing-options {
                      static {
                        route 0.0.0.0/0 next-table inet.0;
                      }
                    }
                  }
                }
```

Routing Instances**Protocols BGP**

```

protocols {
  bgp {
    group to-CE1 {
      import import-public-addr-to-inet0;
      family inet {
        unicast {
          rib-group vpna-to-inet0;
        }
      }
    }
    peer-as 63001;
    neighbor 192.168.197.14;
  }
}

```

Policy Options

```

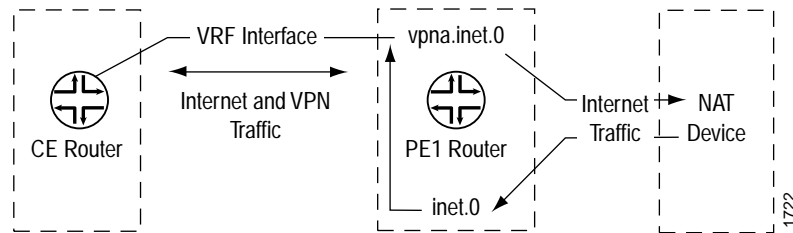
policy-options {
  policy-statement import-public-addr-to-inet0 {
    term a {
      from {
        protocol bgp;
        rib vpna.inet.0;
        community [ public-comm private-comm ];
      }
      then accept;
    }
    term b {
      from {
        protocol bgp;
        community public-comm;
      }
      to rib inet.0;
      then accept;
    }
    term c {
      then reject;
    }
  }
  community private-comm members target:1:333;
  community public-comm members target:1:111;
  community vpna-comm members target:63000:100;
}

```


Route Internet Traffic through a Separate NAT Device

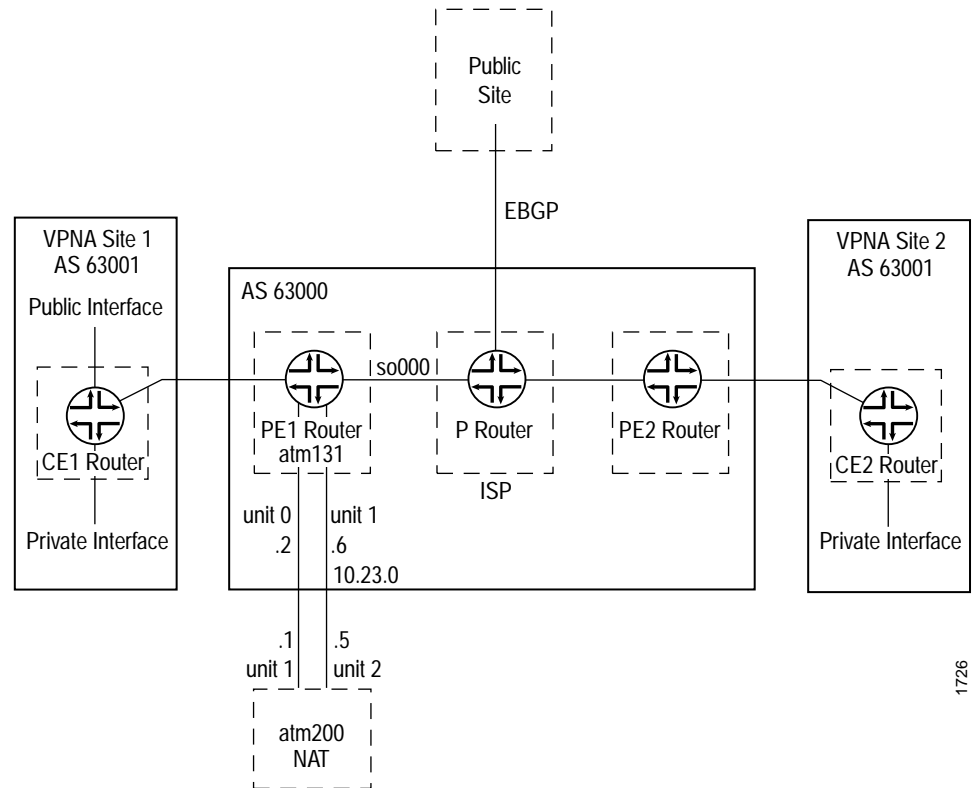
In this example, the CE router does not perform NAT. It sends both VPN and Internet traffic over the same interface to the PE router. The PE router is connected to a NAT device using two interfaces. One interface is configured in the PE router's VRF table and points to a VPN interface on the NAT device, which can route Internet traffic for the VPN. The other interface is in a default instance, for example, part of public routing table `inet.0`. There can be a single physical connection between the PE router and the NAT device and multiple logical connections—one for each VRF table and another interface—as part of the global routing table (see Figure 34).

Figure 34: Overview of Internet Traffic Routed through a Separate NAT Device



This example's topology expands upon that illustrated in Figure 30 on page 218. The CE router sends both VPN and Internet traffic to Router PE1. VPN traffic is routed based on the VPN routes received by Router PE1. Traffic for everything else is sent to the NAT device using Router PE1's private interface to the NAT device, which then translates the private addresses and sends the traffic back to Router PE1 using that router's public interface (see Figure 35).

Figure 35: Internet Traffic Routed through a Separate NAT Device Example Topology



1726

Configure Interfaces on Router PE1

Configure an interface for VPN traffic to and from Router CE1, an interface for VPN traffic to and from the NAT device, and an interface for Internet traffic to and from the NAT device:

```
[edit]
interfaces {
  t3-0/2/0 {
    dce;
    encapsulation frame-relay;
    unit 0 {
      description "to CE1 VPN interface";
      dlci 10;
      family inet {
        address 192.168.197.13/30;
      }
    }
  }
  at-1/3/1 {
    atm-options {
      vpi 1 maximum-vcs 255;
    }
    unit 0 {
      description "to NAT VPN interface";
      vci 1.100;
      family inet {
        address 10.23.0.2/32 {
          destination 10.23.0.1;
        }
      }
    }
    unit 1 {
      description "to NAT public interface";
      vci 1.101;
      family inet {
        address 10.23.0.6/32 {
          destination 10.23.0.5;
        }
      }
    }
  }
}
```

Configure Routing Options for Router PE1

You need to configure a static route on Router PE1 to direct Internet traffic to the CE router through the NAT device. Router PE1 distributes this route to the Internet as follows:

```
[edit]
routing-options {
  static {
    route 10.12.1.0/24 next-hop 10.23.0.5;
  }
}
```

Configure Routing Protocols on Router PE1

Configure MPLS, BGP, IS-IS, and LDP on Router PE1. For the MPLS configuration, include the NAT device's VPN interface in the VRF table. As a part of the BGP configuration, include a policy to advertise the public IP address pool:

```
[edit]
protocols {
  mpls {
    interface t3-0/2/0.0;
    interface at-1/3/1.0;
  }
  bgp {
    group pe-pe {
      type internal;
      local-address 10.255.14.171;
      family inet {
        any;
      }
      family inet-vpn {
        any;
      }
      export [ fix-nh redistribute-static ];
      neighbor 10.255.14.177;
      neighbor 10.255.14.173;
    }
  }
  isis {
    level 1 disable;
    interface so-0/0/0.0;
    interface lo0.0;
  }
  ldp {
    interface so-0/0/0.0;
  }
}
```

Configure a Routing Instance for Router PE1

Configure a routing instance on Router PE1. As part of the routing instance configuration, under routing-options, configure a static default route in vpn0.inet.0 pointing to the NAT device's VPN interface (this directs all non-VPN traffic to the NAT device):

```
[edit]
routing-instances {
  vpn0 {
    instance-type vrf;
    interface t3-0/2/0.0;
    interface at-1/3/1.0;
    route-distinguisher 10.255.14.171:100;
    vrf-import vpn0-import;
    vrf-export vpn0-export;
    routing-options {
      static {
        route 0.0.0.0/0 next-hop 10.23.0.1;
      }
    }
  }
}
```

```

protocols {
    bgp {
        group to-CE1 {
            peer-as 63001;
            neighbor 192.168.197.14;
        }
    }
}
}
policy-options {
    policy-statement fix-nh {
        then {
            next-hop self;
        }
    }
    policy-statement redist-static {
        term a {
            from {
                protocol static;
                route-filter 10.12.1.0/24 exact;
            }
            then accept;
        }
        term b {
            from protocol bgp;
            then accept;
        }
        term c {
            then accept;
        }
    }
    policy-statement vpna-import {
        term a {
            from {
                protocol bgp;
                community vpna-comm;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement vpna-export {
        term a {
            from protocol bgp;
            then {
                community add vpna-comm;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    community vpna-comm members target:63000:100;
}

```

Traffic Routed by Separate NAT Device Configuration Summarized by Router

Router PE1

```

Interfaces  interfaces {
                  t3-0/2/0 {
                      dce;
                      encapsulation frame-relay;
                      unit 0 {
                          description "to CE1 VPN interface";
                          dlci 10;
                          family inet {
                              address 192.168.197.13/30;
                          }
                      }
                  }
                  at-1/3/1 {
                      atm-options {
                          vpi 1 maximum-vcs 255;
                      }
                      unit 0 {
                          description "to NAT VPN interface";
                          vci 1.100;
                          family inet {
                              address 10.23.0.2/32 {
                                  destination 10.23.0.1;
                              }
                          }
                      }
                      unit 1 {
                          description "to NAT public interface";
                          vci 1.101;
                          family inet {
                              address 10.23.0.6/32 {
                                  destination 10.23.0.5;
                              }
                          }
                      }
                  }
              }

Routing Options routing-options {
                      static {
                          route 10.12.1.0/24 next-hop 10.23.0.5;
                      }
                  }
    
```

```

Routing Protocols protocols {
    mpls {
        interface t3-0/2/0.0;
        interface at-1/3/1.0;
    }
    bgp {
        group pe-pe {
            type internal;
            local-address 10.255.14.171;
            family inet {
                any;
            }
            family inet-vpn {
                any;
            }
            export [ fix-nh redist-static ];
            neighbor 10.255.14.177;
            neighbor 10.255.14.173;
        }
    }
    isis {
        level 1 disable;
        interface so-0/0/0.0;
        interface lo0.0;
    }
    ldp {
        interface so-0/0/0.0;
    }
}

Routing Instance routing-instances {
    vpna {
        instance-type vrf;
        interface t3-0/2/0.0;
        interface at-1/3/1.0;
        route-distinguisher 10.255.14.171:100;
        vrf-import vpna-import;
        vrf-export vpna-export;
        routing-options {
            static {
                route 0.0.0.0/0 next-hop 10.23.0.1;
            }
        }
        protocols {
            bgp {
                group to-CE1 {
                    peer-as 63001;
                    neighbor 192.168.197.14;
                }
            }
        }
    }
}

```

```

Policy Options policy-options {
    policy-statement fix-nh {
        then {
            next-hop self;
        }
    }
    policy-statement redist-static {
        term a {
            from {
                protocol static;
                route-filter 10.12.1.0/24 exact;
            }
            then accept;
        }
        term b {
            from protocol bgp;
            then accept;
        }
        term c {
            then accept;
        }
    }
    policy-statement vpna-import {
        term a {
            from {
                protocol bgp;
                community vpna-comm;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement vpna-export {
        term a {
            from protocol bgp;
            then {
                community add vpna-comm;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
    community vpna-comm members target:63000:100;
}

```


Centralized Internet Access

This section describes several ways to configure a CE router to act as a central site for Internet access. Internet traffic from other sites (CE routers) is routed to the hub CE router (which also performs NAT) using its VPN interface. The hub CE router then forwards the traffic to a PE router connected to the Internet through another interface identified in the inet.0 table. The hub CE router can advertise a default route to the spoke CE routers. The disadvantage of this type of configuration is that all traffic has to go through the central CE router before going to the Internet, causing network delays if this router receives too much traffic. However, in a corporate network, traffic might have to be routed to a central site because most corporate networks separate the VPN from the Internet by means of a single firewall.

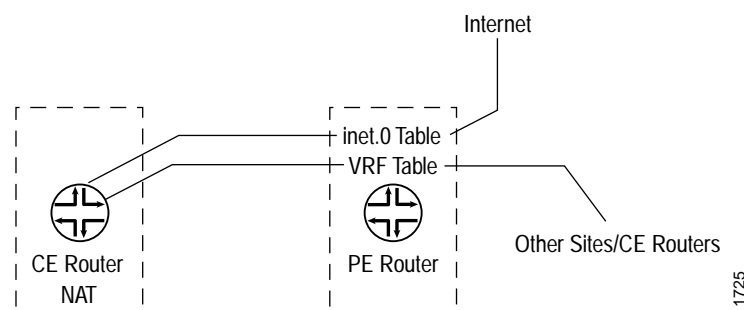
This section includes the following examples:

- Route Internet Traffic through a Hub CE Router on page 241
- Route Internet Traffic through Multiple CE Routers on page 245

Route Internet Traffic through a Hub CE Router

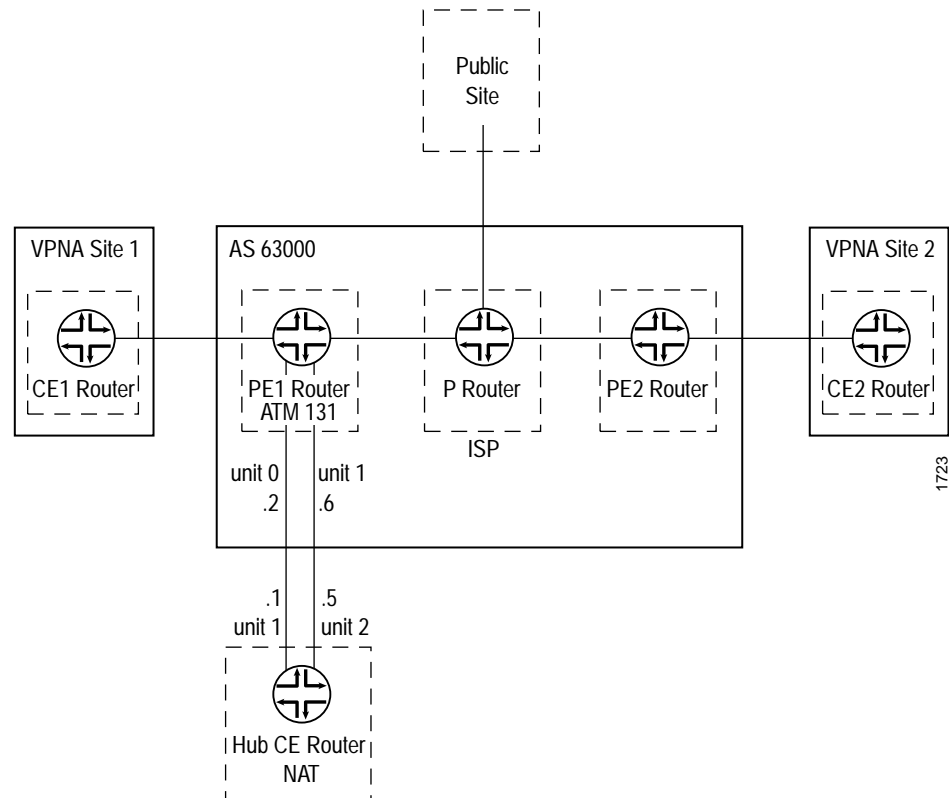
In this example, Internet traffic is routed through a hub CE router. The hub CE router has two interfaces to the hub PE router: a VPN interface and a public interface. It performs NAT on traffic forwarded from the hub PE router through the VPN interface and forwards that traffic from its public interface back to the hub PE router. The hub PE router has a static default route in its VRF table pointing to the hub CE router's VPN interface. It announces this default route to the rest of the VPN, attracting all non-VPN traffic to the hub CE route. The hub PE router also installs and distributes the VPN's public IP address space (see Figure 36).

Figure 36: Internet Access through a Hub CE Router Performing NAT



The configuration for this example is almost identical to that described in “Route Internet Traffic through a Separate NAT Device” on page 233. The difference is that Router PE1 is configured to announce a static default route to the other CE routers (see Figure 37).

Figure 37: Internet Access Provided through a Hub CE Router



Configure a Routing Instance on Router PE1

Configure a routing instance for Router PE1. As part of this configuration, under routing-options, configure a default static route (route 0.0.0.0/0) to be installed in `vpn.inet.0` and point the route to the hub CE router's VPN interface (10.23.0.1). Also, configure BGP under the routing instance to export the default route to the local CE router:

```
[edit]
routing-instances {
  vpn {
    instance-type vrf;
    interface t3-0/2/0.0;
    interface at-1/3/1.0;
    route-distinguisher 10.255.14.171:100;
    vrf-import vpn-import;
    vrf-export vpn-export;
    routing-options {
      static {
        route 0.0.0.0/0 next-hop 10.23.0.1;
      }
    }
  }
}
```

```

protocols {
  bgp {
    group to-CE1 {
      export export-default;
      peer-as 63001;
      neighbor 192.168.197.14;
    }
  }
}

```

Configure Policy Options on Router PE1

Configure policy options on Router PE1. As part of this configuration, Router PE1 should export the static default route to all the remote PE routers in vpna (configured in the policy-statement vpna-export statement under term b):

```

[edit]
policy-options {
  policy-statement vpna-export {
    term a {
      from protocol bgp;
      then {
        community add vpna-comm;
        accept;
      }
    }
    term b {
      from {
        protocol static;
        route-filter 0.0.0.0/0 exact;
      }
      then {
        community add vpna-comm;
        accept;
      }
    }
    term c {
      then reject;
    }
  }
  policy-statement export-default {
    term a {
      from {
        protocol static;
        route-filter 0.0.0.0/0 exact;
      }
      then accept;
    }
    term b {
      from protocol bgp;
      then accept;
    }
    term c {
      then reject;
    }
  }
}

```

Internet Traffic Routed by a Hub CE Router Configuration Summarized by Router

Router PE1

The configuration for Router PE1 is almost identical to that for the example in “Route Internet Traffic through a Separate NAT Device” on page 233. The difference is that Router PE1 is configured to announce a static default route to the other CE routers.

```

Routing Instance routing-instances {
  vpn {
    instance-type vrf;
    interface t3-0/2/0.0;
    interface at-1/3/1.0;
    route-distinguisher 10.255.14.171:100;
    vrf-import vpn-import;
    vrf-export vpn-export;
    routing-options {
      static {
        route 0.0.0.0/0 next-hop 10.23.0.1;
      }
    }
    protocols {
      bgp {
        group to-CE1 {
          export export-default;
          peer-as 63001;
          neighbor 192.168.197.14;
        }
      }
    }
  }
}

```

```

Policy Options policy-options {
  policy-statement vpn-export {
    term a {
      from protocol bgp;
      then {
        community add vpn-comm;
        accept;
      }
    }
    term b {
      from {
        protocol static;
        route-filter 0.0.0.0/0 exact;
      }
      then {
        community add vpn-comm;
        accept;
      }
    }
    term c {
      then reject;
    }
  }
}

```

```

policy-statement export-default {
  term a {
    from {
      protocol static;
      route-filter 0.0.0.0/0 exact;
    }
    then accept;
  }
  term b {
    from protocol bgp;
    then accept;
  }
  term c {
    then reject;
  }
}

```

Route Internet Traffic through Multiple CE Routers

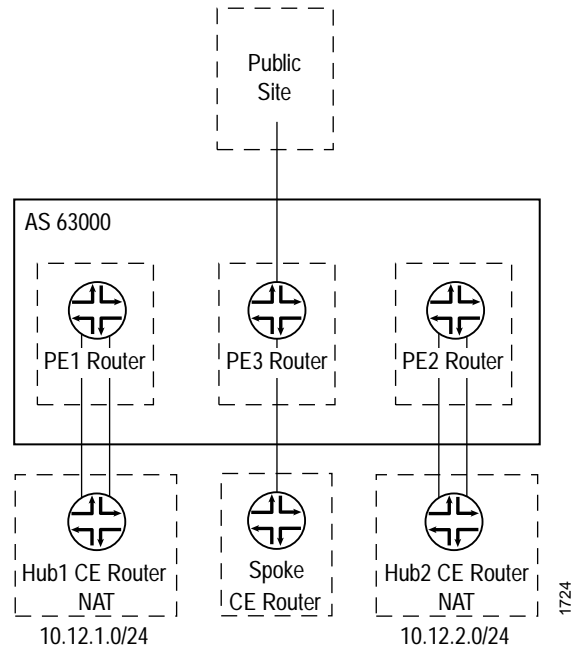
The example in this section is an extension of that described in “Route Internet Traffic through a Hub CE Router” on page 241. This example provides different exit points for different sites by using multiple hub CE routers performing similar functions. Each hub CE router tags the default route with a different route target and allows the spoke CE routers to select the hub site that should be used for Internet access (see Figure 38).

This example uses two hub CE routers that handle NAT and Internet traffic:

- Hub1 CE router tags 0/0 with community public-comm1 (target: 1:111)
- Hub2 CE router tags 0/0 with community public-comm2 (target: 1:112)

Note that the spoke CE router in this example is configured to have a bias toward Hub2 for Internet access.

Figure 38: Two Hub CE Routers Handling Internet Traffic and NAT



Configure a Routing Instance on Router PE1

Configure a routing instance on Router PE1 as follows:

```
[edit]
routing-instances {
  vpna {
    instance-type vrf;
    interface t3-0/2/0.0;
    interface at-1/3/1.0;
    route-distinguisher 10.255.14.171:100;
    vrf-import vpna-import;
    vrf-export vpna-export;
    routing-options {
      static {
        route 0.0.0.0/0 next-hop 10.23.0.1;
      }
    }
  }
  protocols {
    bgp {
      group to-CE1 {
        export export-default;
        peer-as 63001;
        neighbor 192.168.197.14;
      }
    }
  }
}
```

Configure Policy Options on Router PE1

The policy options for Router PE1 are the same as in “Route Internet Traffic through a Hub CE Router” on page 241, but the configuration in this example includes an additional community, public-comm1, in the export statement:

```
[edit]
policy-options {
  policy-statement vpna-import {
    term a {
      from {
        protocol bgp;
        community vpna-comm;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement vpna-export {
    term a {
      from {
        protocol static;
        route-filter 0.0.0.0/0 exact;
      }
      then {
        community add public-comm1;
        community add vpna-comm;
        accept;
      }
    }
    term b {
      from protocol bgp;
      then {
        community add vpna-comm;
        accept;
      }
    }
    term c {
      then reject;
    }
  }
  community public-comm1 members target:1:111;
  community public-comm2 members target:1:112;
  community vpna-comm members target:63000:100;
}
```

Note that the configuration of Router PE2 is the identical to that of Router PE1 except that Router PE2 exports the default route through community public-comm2.

Configure a Routing Instance on Router PE3

Configure routing instance vpna on Router PE3 as follows:

```
[edit]
routing-instances {
  vpna {
    instance-type vrf;
    interface t1-0/2/0.0;
    route-distinguisher 10.255.14.173:100;
    vrf-import vpna-import;
    vrf-export vpna-export;
    protocols {
      rip {
        group to-vpn12 {
          export export-CE;
          neighbor t1-0/2/0.0;
        }
      }
    }
  }
}
```

Configure Policy Options on Router PE3

The vrf-import policy for Router PE3 is configured to select the Internet exit point based on the additional communities specified in “Configure Policy Options on Router PE1” on page 247:

```
[edit]
policy-options {
  policy-statement vpna-export {
    term a {
      from protocol rip;
      then {
        community add vpna-comm;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  policy-statement vpna-import {
    term a {
      from {
        protocol bgp;
        community public-comm1;
        route-filter 0.0.0.0/0 exact;
      }
      then reject;
    }
    term b {
      from {
        protocol bgp;
        community vpna-comm;
      }
      then accept;
    }
  }
}
```



```

        term c {
            then reject;
        }
    }
    policy-statement export-CE {
        from protocol bgp;
        then accept;
    }
    community vpna-comm members target:69:100;
    community public-comm1 members target:1:111;
    community public-comm2 members target:1:112;
}

```

Route Internet Traffic through Multiple CE Routers Configuration Summarized by Router

Router PE1

This configuration is an extension of the example in “Route Internet Traffic through a Hub CE Router” on page 241. It provides different exit points for various sites by using multiple hub CE routers that perform similar functions.

```

Routing Instance  routing-instances {
                    vpna {
                        instance-type vrf;
                        interface t3-0/2/0.0;
                        interface at-1/3/1.0;
                        route-distinguisher 10.255.14.171:100;
                        vrf-import vpna-import;
                        vrf-export vpna-export;
                        routing-options {
                            static {
                                route 0.0.0.0/0 next-hop 10.23.0.1;
                            }
                        }
                    }
                    protocols {
                        bgp {
                            group to-CE1 {
                                export export-default;
                                peer-as 63001;
                                neighbor 192.168.197.14;
                            }
                        }
                    }
                }

Policy Options    policy-options {
                    policy-statement vpna-import {
                        term a {
                            from {
                                protocol bgp;
                                community vpna-comm;
                            }
                            then accept;
                        }
                        term b {
                            then reject;
                        }
                    }
                }

```

```

policy-statement vpn-export {
  term a {
    from {
      protocol static;
      route-filter 0.0.0.0/0 exact;
    }
    then {
      community add public-comm1;
      community add vpn-comm;
      accept;
    }
  }
  term b {
    from protocol bgp;
    then {
      community add vpn-comm;
      accept;
    }
  }
  term c {
    then reject;
  }
}
community public-comm1 members target:1:111;
community public-comm2 members target:1:112;
community vpn-comm members target:63000:100;
}

```

Router PE2

The configuration of Router PE2 is the identical to that of Router PE1, except that Router PE2 exports the default route through community public-comm2 (see “Policy Options” on page 249).

Router PE3

```

Routing Instances routing-instances {
  vpn {
    instance-type vrf;
    interface t1-0/2/0.0;
    route-distinguisher 10.255.14.173:100;
    vrf-import vpn-import;
    vrf-export vpn-export;
    protocols {
      rip {
        group to-vpn12 {
          export export-CE;
          neighbor t1-0/2/0.0;
        }
      }
    }
  }
}

```

```

Policy Options  policy-options {
                    policy-statement vpna-export {
                      term a {
                        from protocol rip;
                        then {
                          community add vpna-comm;
                          accept;
                        }
                      }
                      term b {
                        then reject;
                      }
                    }
                    policy-statement vpna-import {
                      term a {
                        from {
                          protocol bgp;
                          community public-comm1;
                          route-filter 0.0.0.0/0 exact;
                        }
                        then reject;
                      }
                      term b {
                        from {
                          protocol bgp;
                          community vpna-comm;
                        }
                        then accept;
                      }
                      term c {
                        then reject;
                      }
                    }
                    policy-statement export-CE {
                      from protocol bgp;
                      then accept;
                    }
                    community vpna-comm members target:69:100;
                    community public-comm1 members target:1:111;
                    community public-comm2 members target:1:112;
                  }

```

.....

Chapter 11

Summary of Layer 3 VPN Configuration Statements

The following sections explain the major routing-instances configuration statements that apply specifically to Layer 3 virtual private networks (VPNs). The statements are organized alphabetically. Routing instances and the statements at the [edit routing-instances *routing-instance-name* routing-options] and [edit routing-instances *routing-instance-name* protocols] hierarchy levels are explained in the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

description

Syntax	description <i>text</i> ;
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i>]
Description	Allows you to provide a textual description for the routing instance. Enclose any descriptive text that includes spaces in quotation marks (" "). Any descriptive text you include is displayed in the output of the show route instance detail command and has no effect on the operation of the routing instance.
Usage Guidelines	See "Configure the Description" on page 86.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

inet6-vpn

Syntax	inet6-vpn {unicast multicast any } {...}
Hierarchy Level	[edit protocols bgp group <i>group-name</i> family]
Description	Enables IPv6 on the PE router for the Layer 3 VPN.
Options	prefix-limit maximum—Specify the maximum prefix limit. The value can be from 1 to 4,294,967,295. rib-group—Specify the name of the routing table group.
Usage Guidelines	See "Configure IPv6 between the PE and CE Routers" on page 102.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

instance-type

Syntax	instance-type vrf;
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i>]
Description	Defines the type of routing instance.
Options	vrf—VPN routing and forwarding instance. Required to create a VPN. Creates a VPN routing and forwarding (VRF) table (<i>instance-name.inet.0</i>), which contains the routes originating from and destined for a particular VPN. You must configure the interface, route-distinguisher, vrf-import, and vrf-export statements for this type of routing instance.
Usage Guidelines	See “Configure the Instance Type” on page 86.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

interface

Syntax	interface <i>interface-name</i> ;
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i>]
Description	Interface over which the VPN traffic travels between the provider edge (PE) router and customer edge (CE) router. You configure the interface on the PE router. If the instance type is vrf, the interface statement is required.
Usage Guidelines	See “Configure Interfaces for VPN Routing” on page 86.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

route-distinguisher

Syntax	route-distinguisher (<i>as-number:number</i> <i>ip-address:number</i>);
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i>]
Description	<p>Identifier attached to routes that distinguishes to which VPN it belongs. Each routing instance must have a unique route distinguisher associated with it. If the instance type is vrf, the route-distinguisher statement is required.</p> <p>If you configure the route-distinguisher-id statement at the [edit routing-options] hierarchy level, a type 1 route distinguisher is automatically assigned to VRF routing instances. For more information, see the <i>JUNOS Internet Software Configuration Guide: Routing and Routing Protocols</i>.</p> <p>The route distinguisher is a 6-byte value that you can specify in one of the following formats:</p> <ul style="list-style-type: none"> ■ <i>as-number:number</i>, where <i>as-number</i> is your assigned autonomous system (AS) number (a 2-byte value) and <i>number</i> is any 4-byte value. The AS number can be in the range of 1 through 65,535. ■ <i>ip-address:number</i>, where <i>ip-address</i> is an IP address in your assigned prefix range (a 4-byte value) and <i>number</i> is any 2-byte value. The IP address can be any globally unique unicast address.
Usage Guidelines	See “Configure the Route Distinguisher” on page 21.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

route-distinguisher-id

Syntax	route-distinguisher-id <i>ip-address</i> ;
Hierarchy Level	[edit routing-options]
Description	<p>When configured, a route distinguisher is automatically assigned to the routing instance. If you configure the route-distinguisher statement in addition to the route-distinguisher-id statement, the value configured for route-distinguisher supersedes the value generated from route-distinguisher-id.</p>
Usage Guidelines	See “Configure the Route Distinguisher” on page 21.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

vpn-apply-export

Syntax	vpn-apply-export;
Hierarchy Level	[edit protocols bgp] [edit protocols bgp group <i>group-name</i>] [edit protocols bgp group <i>group-name</i> neighbor <i>neighbor</i>]
Description	Applies both the VRF export and BGP group or neighbor export policies (VRF first, then BGP) before routes are advertised in the vrf or l2vpn routing tables to other PE routers.
Usage Guidelines	See “Apply Both the VRF Export and the BGP Export Policies” on page 92.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

vrf-export

Syntax	vrf-export [<i>policy-names</i>];
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i>]
Description	How routes are exported from the local PE router’s VRF table (<i>routing-instance-name</i> .inet.0) to the remote PE router. If the instance type is vrf, the vrf-export statement is required.
Options	You can configure multiple export policies on the PE router.
Usage Guidelines	See “Configure Export Policy for the PE Router’s VRF Table” on page 91.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

vrf-import

Syntax	vrf-import [<i>policy-names</i>];
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i>]
Description	How routes are imported into the local PE router’s VRF table (<i>routing-instance-name</i> .inet.0) from the remote PE router. If the instance type is vrf, the vrf-import statement is required.
Options	You can configure multiple import policies on the PE router.
Usage Guidelines	See “Configure Import Policy for the PE Router’s VRF Table” on page 90.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

vrf-table-label

Syntax	vrf-table-label;
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i>]
Description	Makes it possible to map the inner label of a packet to a specific VRF and thus allows the examination of the encapsulated IP header.
Usage Guidelines	See “Filter Traffic Based on the IP Header” on page 94.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

vrf-target

Syntax	vrf-target;
Hierarchy Level	[edit routing-instances <i>routing-instance-name</i>]
Description	Configure a single policy for import and a single policy for export to replace the per-VRF policies for every community.
Options	import—Specifies the allowed communities to accept from neighbors. export—Specifies the allowed communities to send to neighbors.
Usage Guidelines	See “Configure a VRF Target” on page 93.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

Part 4

Interprovider and Carrier-of-Carriers VPNs

- Interprovider and Carrier-of-Carriers VPNs Overview on page 261
- Interprovider and Carrier-of-Carriers Configuration Guidelines on page 269
- Configuration Examples for Interprovider and Carrier-of-Carriers VPNs on page 287
- Summary of the Interprovider and Carrier-of-Carriers Configuration Statement on page 321

.....

Chapter 12

Interprovider and Carrier-of-Carriers VPNs Overview

This chapter describes in detail the operation of interprovider and carrier-of-carriers virtual private networks (VPNs) as described in *BGP/MPLS VPNs*, RFC 2547bis. As VPNs are deployed on the Internet, the customer of a VPN service provider might be another service provider rather than an end customer. The customer service provider depends on the VPN service provider to deliver a VPN transport service between the customer service provider's points of presence (POPs) or regional networks.

If the customer service provider's sites have different autonomous system (AS) numbers, then the VPN transit service provider supports carrier-of-carrier VPN service for the inter-provider VPN service. If the customer service provider's sites have the same AS number, then the VPN transit service provider delivers a carrier-of-carriers VPN service.

This chapter includes the following sections:

- Interprovider and Carrier-of-Carriers VPN Standards on page 261
- Traditional VPNs, Interprovider VPNs, and Carrier-of-Carriers VPNs on page 262
- Interprovider VPNs on page 263
- Carrier-of-Carriers VPNs on page 265

Interprovider and Carrier-of-Carriers VPN Standards

Interprovider and carrier-of-carriers VPNs are defined by the following documents:

- RFC 3107, *Carrying Label Information in BGP-4*.
- *BGP/MPLS VPNs*, Internet draft draft-ietf-ppvpn-rfc2547bis-00.txt.
- *RFC 2547bis: BGP/MPLS VPN Fundamentals*—a white paper located on the Juniper Web site at <http://www.juniper.net/>.

To access Internet RFCs and drafts, go to the Internet Engineering Task Force (IETF) Web site at <http://www.ietf.org/>.

Traditional VPNs, Interprovider VPNs, and Carrier-of-Carriers VPNs

The sections that follow provide an overview of traditional VPNs, interprovider and carrier-of-carriers VPNs, and the differences in how external and internal routes are handled in each of these environments.

In traditional IP routing architectures, there is a clear distinction between internal routes and external routes. From the perspective of an Internet service provider (ISP), internal routes include all the provider's internal links (including Border Gateway Protocol [BGP] next-hops) and loopback interfaces. These internal routes are exchanged with other routers in the ISP's network by means of an interior gateway protocol (IGP) such as Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS). All routes learned at Internet peering points or from customer sites are classified as external routes and are distributed using an exterior gateway protocol (EGP) such as BGP. In traditional IP routing architectures, the number of internal routes is typically much smaller than the number of external routes.

Standard VPNs

The traditional distinction between internal routes and external routes also applies to VPN routing architectures. As shown in Figure 1 on page 4, the provider (P) routers maintain only the service provider's internal routes (to PE routers and other P routers); they do not maintain VPN routes. PE routers are the only devices in the provider network that are required to maintain external routes.

The BGP next-hop connects the external routes to the internal routes in traditional VPNs:

- The BGP next-hop is advertised with each external route in BGP advertisements.
- The route to the BGP next-hop is an internal route that is advertised by the IGP.
- Multiprotocol Label Switching (MPLS) provides packet forwarding from the ingress PE router to the BGP next-hop egress PE router.

Interprovider and Carrier-of-Carriers VPNs

All interprovider and carrier-of-carriers VPNs share the following characteristics:

- Each interprovider or carrier-of-carriers VPN customer must distinguish between internal and external customer routes.
- Internal customer routes must be maintained by the VPN service provider in its PE routers.
- External customer routes are carried only by the customer's routers, not by the VPN service provider's routers.

The key difference between interprovider and carrier-of-carriers VPNs is whether the customer sites belong to the same AS or to separate ASs:

- **Interprovider VPNs**—The customer sites belong to different ASs. You need to configure external Border Gateway Protocol (EBGP) to exchange the customer's external routes.
- **Carrier-of-Carriers VPNs**—The customer sites belong to the same AS. You need to configure internal Border Gateway Protocol (IBGP) to exchange the customer's external routes.

In general, each service provider in a VPN hierarchy is required to maintain its own internal routes in its provider routers, and the internal routes of its customers in its PE routers. By recursively applying this rule, it is possible to create a hierarchy of VPNs.

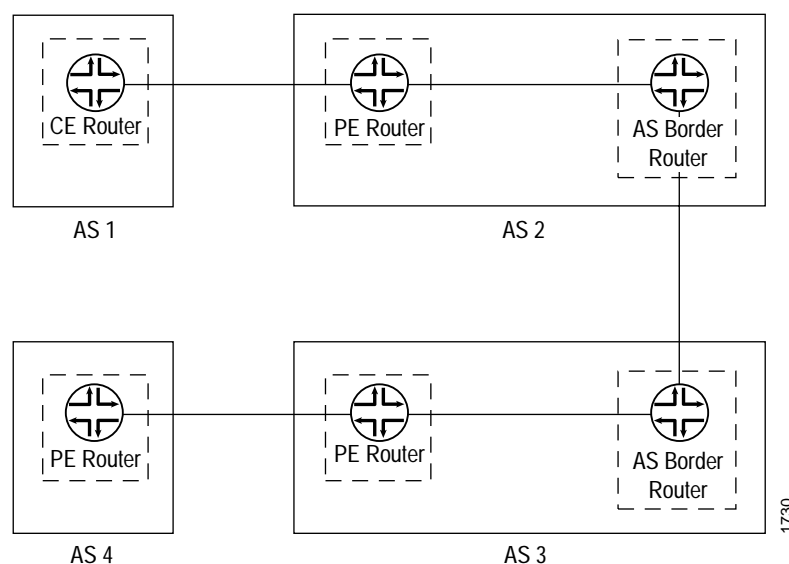
The following are definitions of the types of PE routers specific to interprovider and carrier-of-carriers VPNs:

- The AS border router is located at the AS border and handles traffic leaving and entering the AS.
- The end-PE router is the PE router in the customer VPN; it is connected to the CE router at the end customer's site.

Interprovider VPNs

Interprovider VPNs provide connectivity between separate ASs. This functionality might be used by a VPN customer who has connections to several different ISPs, or different connections to the same ISP in different geographic regions, each of which has a different AS. Figure 39 illustrates the type of network topology used by an interprovider VPN.

Figure 39: Interprovider VPN Network Topology



The following sections describe the ways you can configure an interprovider VPN:

- Interprovider VPNs—Linking VRF Tables between Autonomous Systems on page 264
- Interprovider VPNs—Configure MP-EBGP between AS Border Routers on page 264
- Interprovider VPNs—Configure Multihop MP-EBGP on page 265

Interprovider VPNs—Linking VRF Tables between Autonomous Systems

You can connect two separate ASs by simply linking the VRF table in the AS border router of one AS to the VRF table in the AS border router in the other AS. Each AS border router must contain a VRF instance for every VPN configured in both service provider networks. You then configure an IP session between the two AS border routers. In effect, the AS border routers treat each other as CE routers.

Because of the complexity of the configuration, particularly with regard to scaling, this method is not recommended. The details of this configuration are not provided in this manual.

Interprovider VPNs—Configure MP-EBGP between AS Border Routers

In this approach, the PE routers within an AS use multiprotocol external BGP (MP-EBGP) to distribute labeled VPN-Internet Protocol version 4 (IPv4) routes to an AS border router or a route reflector of which the AS border router is a client. The AS border router uses multiprotocol external BGP (MP-EBGP) to distribute the labeled VPN-IPv4 routes to its peer AS border router in the neighboring AS. The peer AS border router then uses MP-IBGP to distribute labeled VPN-IPv4 routes to PE routers, or to a route reflector of which the PE routers are a client.

This approach enhances the scalability of an EBGp VRF-to-VRF configuration because it eliminates the need to configure all of the VPNs on every AS border router. However, it also introduces some complexity:

- All the VRF routes must be stored in the AS border router.
- An LSP must be established from ingress PE routers to egress PE routers.
- Secure connections must exist among the ASs along the path from the ingress PE router to the egress PE router.
- The ASs must be configured to know which AS border routers receive routes with specific route target attributes.

Interprovider VPNs—Configure Multihop MP-EBGP

In this type of interprovider VPN configuration, provider routers do not need to know all the routes in all the VPNs. Only the PE routers must have all the VPN routes. The provider routers simply forward traffic to the PE routers—they are not aware of the packets' destination. The connections between the AS border routers in separate ASs forward traffic between the ASs, much as a label-switched path (LSP) works.

The following are the basic steps you take to configure an interprovider VPN in this manner:

1. Configure multihop EBGp redistribution of labeled VPN-IPv4 routes between the source and destination ASs.
2. Configure EBGp to redistribute labeled IPv4 routes from its AS to neighboring ASs.
3. Configure MPLS on the end-PE routers of the VPNs.

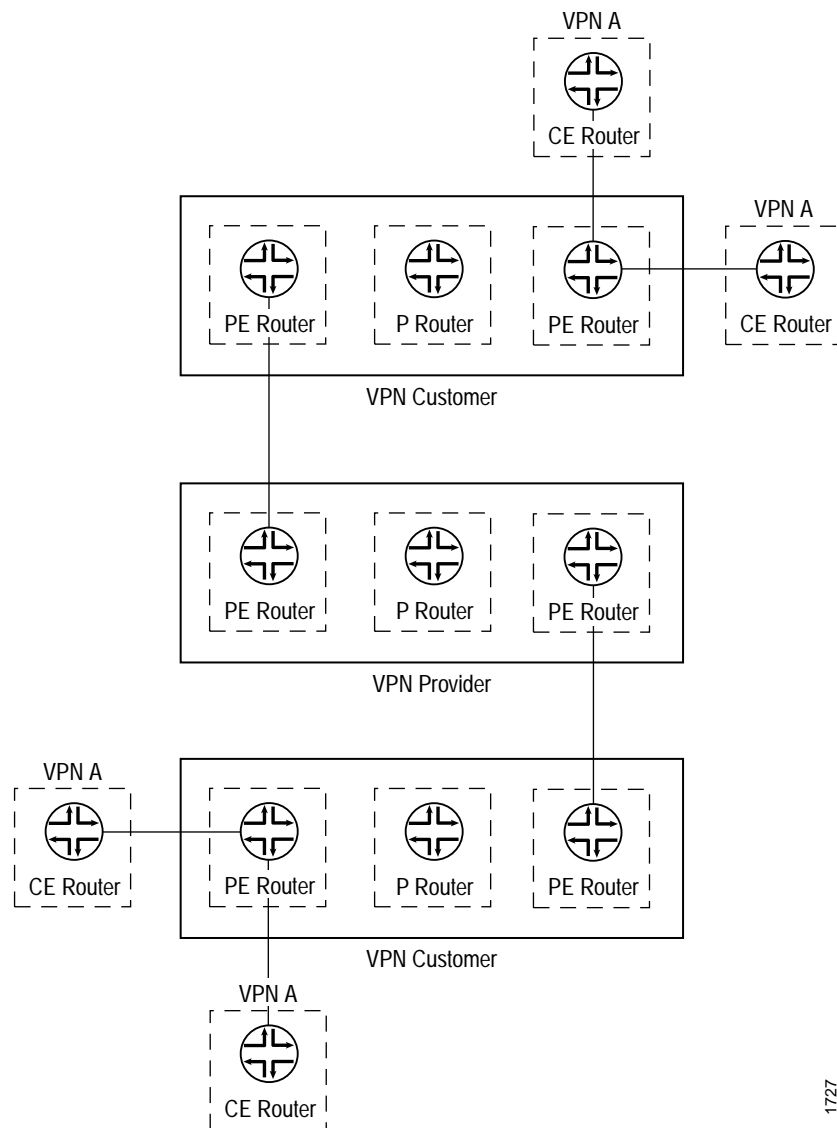
Carrier-of-Carriers VPNs

The customer of a VPN service provider might be a service provider for the end customer. The following are the two main types of carrier-of-carriers VPNs (as described in *BGP/MPLS VPNs*):

- Internet Service Provider as the Customer on page 267—The VPN customer is an ISP that uses the VPN service provider's network to connect its geographically disparate regional networks. The customer does not have to configure MPLS within its regional networks.
- VPN Service Provider as the Customer on page 267—The VPN customer is itself a VPN service provider offering VPN service to its customers. The carrier-of-carriers VPN service customer relies on the backbone VPN service provider for intersite connectivity. The customer VPN service provider is required to run MPLS within its regional networks.

Figure 40 illustrates the network architecture used for a carrier-of-carriers VPN service.

Figure 40: Carrier-of-Carriers VPN Architecture



Internet Service Provider as the Customer

In this type of carrier-of-carriers VPN configuration, ISP A configures its network to provide Internet service to ISP B. ISP B provides the connection to the customer wanting Internet service, but the actual Internet service is provided by ISP A.

This type of carrier-of-carriers VPN configuration has the following characteristics:

- The carrier-of-carriers VPN service customer (ISP B) does not need to configure MPLS on its network.
- The carrier-of-carriers VPN service provider (ISP A) must configure MPLS on its network.
- MPLS must also be configured on the CE routers and PE routers connected together in the carrier-of-carriers VPN service customer's and carrier-of-carriers VPN service provider's networks.

VPN Service Provider as the Customer

A VPN service provider can have customers that are themselves VPN service providers. In this type of configuration, also called a hierarchical or recursive VPN, the customer VPN service provider's VPN-IPv4 routes are considered external routes, and the backbone VPN service provider does not import them into its VRF table. The backbone VPN service provider only imports the customer VPN service provider's internal routes into its VRF.

This type of configuration is similar to the configuration described in the "Internet Service Provider as the Customer" section. The similarities and differences are shown in Table 2.

Table 2: Comparison of Interprovider and Carrier-of-Carriers VPNs

Feature	ISP Customer	VPN Service Provider Customer
Customer edge device	AS border router	PE router
IBGP sessions	Carry IPv4 routes	Carry external VPN-IPv4 routes with associated labels
Forwarding within the customer network	MPLS is optional	MPLS is required

.....

Chapter 13

Interprovider and Carrier-of-Carriers Configuration Guidelines

To configure interprovider or carrier-of-carriers virtual private network (VPN) functionality, you typically need to include the `labeled-unicast` statement in the configuration for the Border Gateway Protocol (BGP) on the autonomous system (AS) border routers of an interprovider VPN or the provider edge (PE) and customer edge (CE) routers of a carrier-of-carriers VPN. You must also configure the provider (P) routers in the service provider's and service customer's networks.

To configure interprovider or carrier-of-carriers VPN functionality, you include statements at the `[edit protocols bgp]` hierarchy level:

```
[edit]
protocols {
  bgp {
    group group-name {
      type internal;
      local-address address;
      family inet {
        labeled-unicast;
      }
      neighbor address;
    }
  }
}
```

This chapter contains the following sections:

- Interprovider VPNs on page 269
- Carrier-of-Carriers VPNs on page 274

Interprovider VPNs

You can configure interprovider VPN service in one of the following ways:

- Interprovider VPNs Using MP-EBGP on page 270
- Interprovider VPNs Using Multihop MP-EBGP on page 272

Interprovider VPNs Using MP-EBGP

To configure interprovider VPN service using multiprotocol external BGP (MP-EBGP), you need to configure the AS border routers of each AS. See Figure 39, “Interprovider VPN Network Topology,” on page 263 for an illustration of how the routers interconnect in an interprovider VPN service.

Configure the AS Border Routers

The configuration of the AS border routers in each AS is nearly identical. You need to configure the following on each AS border router:

- Configure RSVP on page 270
- Configure MPLS on page 270
- Configure BGP on page 271
- Configure OSPF on page 271

Configure RSVP

Configure an interface for VPN traffic from the other AS to the PE router handling VPN traffic in the current AS at the [edit protocols rsvp] hierarchy level:

```
[edit]
protocols {
  rsvp {
    interface interface-name;
  }
}
```

Configure MPLS

Configure a label-switched path (LSP) to the PE router at the [edit protocols mpls] hierarchy level. Also configure the interfaces handling VPN traffic from the other AS and to the PE router in the current AS:

```
[edit]
protocols {
  mpls {
    label-switched-path path-name {
      to address;
    }
    interface interface-name;
    interface interface-name;
  }
}
```

Configure BGP

Configure an MP-EBGP session on the AS border router. This session exchanges VPN-Internet Protocol version 4 (IPv4) routes with the AS border router in the other AS.

Configure the MP-EBGP session at the [edit protocols bgp] hierarchy level. Configure a group to handle internal BGP (IBGP) and a group to handle external BGP (EBGP):

```
[edit]
protocols {
  bgp {
    keep all;
    group group-name;
    type internal;
    local-address address;
    family inet-vpn {
      unicast;
    }
    neighbor address;
  }
  group group-name {
    type external;
    family inet-vpn {
      unicast;
    }
    neighbor address {
      peer-as as number;
    }
  }
}
```

Configure OSPF

Configure Open Shortest Path First (OSPF) on the AS border router at the [edit protocols ospf] hierarchy level as follows:

```
[edit]
protocols {
  ospf {
    traffic engineering;
    area address {
      interface interface-name;
      interface interface-name {
        passive;
      }
    }
  }
}
```

Interprovider VPNs Using Multihop MP-EBGP

This section describes how to configure a network to provide interprovider VPN service using multihop MP-EBGP. With this type of configuration, you need to set up the AS border routers and the PE routers connected to the end customer's CE routers. See Figure 39, "Interprovider VPN Network Topology," on page 263 for an illustration of how the routers interconnect in an interprovider VPN service.

Configure the AS Border Routers

The configuration of the AS border routers in each AS is nearly identical. You need to configure the following on each AS border router:

- Configure BGP on page 272
- Configure Policy Options on page 273

Configure BGP

Configure BGP on the AS border routers at the [edit protocols bgp] hierarchy level. Configure a group for IBGP and a group for EBGP to the PE router as follows:

```
[edit]
protocols {
  bgp {
    group group-name {
      type internal;
      local-address address;
      family inet {
        labeled-unicast {
          resolve-vpn;
        }
      }
      neighbor address;
    }
    group group-name {
      type external;
      family inet {
        labeled-unicast;
      }
      export internal;
      neighbor address {
        peer-as as-number;
      }
    }
  }
}
```


Configure Policy Options

Configure the policy options on the AS border routers as follows at the [edit policy-options] hierarchy level:

```
[edit]
policy-options {
  policy-statement policy-name {
    term term-name {
      from protocol [ospf direct];
      then accept;
    }
    term term-name {
      then reject;
    }
  }
}
```

Configure the PE Router

You need to configure a multihop MP-EBGP session on the PE router connected to the end customer's CE router.

Include the labeled-unicast statement at the [edit protocols bgp group family inet] hierarchy level to pass labeled IPv4 routes:

```
[edit]
protocols {
  bgp {
    group group-name {
      type internal;
      local-address address;
      family inet {
        labeled-unicast {
          resolve-vpn;
        }
      }
      neighbor address;
    }
  }
}
```

Configure a group at the [edit protocols bgp] hierarchy level to handle an EBGp multihop session with the remote PE router (to pass VPN-IPv4 routes):

```
[edit]
protocols {
  bgp {
    group group-name {
      multihop {
        ttl 10;
      }
      family inet-vpn {
        unicast;
      }
      neighbor address {
        peer-as as-number;
      }
    }
  }
}
```

Carrier-of-Carriers VPNs

You can configure carrier-of-carriers VPN service in one of the following ways:

- Carrier-of-Carriers VPN—Customer Provides Internet Service on page 274
- Carrier-of-Carriers VPN—Customer Provides VPN Service on page 278

Carrier-of-Carriers VPN—Customer Provides Internet Service

In this type of carrier-of-carriers VPN service configuration, the customer provides basic Internet service. The carrier-of-carriers VPN service provider must configure Multiprotocol Label Switching (MPLS) in its network, although this is optional for the carrier service customer. Figure 40, “Carrier-of-Carriers VPN Architecture,” on page 266 shows how the routers in this type of service interconnect.

This section describes the following:

- Configure the Carrier-of-Carriers VPN Service Customer’s CE Router on page 274
- Configure the Carrier-of-Carriers VPN Service Provider’s PE Routers on page 276

Configure the Carrier-of-Carriers VPN Service Customer’s CE Router

The carrier-of-carriers VPN service customer’s router acts as a CE router with respect to the service provider’s PE router. This section describes how to configure the carrier-of-carriers VPN service customer’s CE router.

Configure MPLS

Configure MPLS at the [edit protocols mpls] hierarchy level as follows on the customer’s CE router:

```
[edit]
protocols {
  mpls {
    traffic-engineering bgp-igp;
    interface interface-name;
  }
}
```

Configure BGP

Configure a group at the [edit protocols bgp] hierarchy level to collate the customer's internal routes:

```
[edit]
protocols {
  bgp {
    group group-name {
      type internal;
      local-address address;
      neighbor address;
    }
  }
}
```

The customer's CE router must be able to send labels to the VPN service provider's router. Enable this by including the labeled-unicast statement at the [edit protocols bgp group neighbor family inet] hierarchy level:

```
[edit]
protocols {
  bgp {
    group group-name {
      export internal;
      peer-as as-number;
      neighbor address {
        family inet {
          labeled-unicast;
        }
      }
    }
  }
}
```

Configure OSPF

Configure OSPF at the [edit protocols ospf] hierarchy level on the customer's CE router as follows:

```
[edit]
protocols {
  ospf {
    area area-id {
      interface interface-name {
        passive;
      }
      interface interface-name;
    }
  }
}
```

Configure Policy Options

Configure policy options at the [edit policy-options] hierarchy level on the customer's CE router as follows:

```
[edit]
policy-options {
  policy-statement statement-name {
    term term-name {
      from protocol [ospf direct ldp];
      then accept;
    }
    term term-name {
      then reject;
    }
  }
}
```

Configure the Carrier-of-Carriers VPN Service Provider's PE Routers

The service provider's PE routers connect to the customer's CE routers and forward the customer's VPN traffic across the provider's network.

Configure MPLS

Configure MPLS at the [edit protocols mpls] hierarchy level on the provider's PE routers as follows:

```
[edit]
protocols {
  mpls {
    interface interface-name;
    interface interface-name;
  }
}
```

Configure BGP

Configure a BGP session at the [edit protocols bgp] hierarchy level with the provider PE router at the other end of the provider's network as follows:

```
[edit]
protocols {
  bgp {
    group group-name {
      type internal;
      local-address address;
      family inet-vpn {
        any;
      }
      neighbor address;
    }
  }
}
```

Configure IS-IS

Configure Intermediate System-to-Intermediate System (IS-IS) at the [edit protocols isis] hierarchy level on the provider's PE routers as follows:

```
[edit]
protocols {
  isis {
    interface interface-name;
    interface interface-name {
      passive;
    }
  }
}
```

Configure LDP

Configure the Label Distribution Protocol (LDP) at the [edit protocols ldp] hierarchy level on the provider's PE routers as follows:

```
[edit]
protocols {
  ldp {
    interface interface-name;
  }
}
```

Configure a Routing Instance

At the [edit routing-instances] hierarchy level, configure layer 3 VPN service with the customer's CE router. Note that you include the `labeled-unicast` statement within the routing instance so the PE router can send labels to the customer's CE router.

```
[edit]
routing-instances {
  routing-instance-name {
    instance-type vrf;
    interface interface-name;
    route-distinguisher address;
    vrf-import policy-name;
    vrf-export policy-name;
    protocols {
      bgp {
        group group-name {
          peer-as as-number;
          neighbor address {
            family inet {
              labeled-unicast;
            }
          }
        }
      }
    }
  }
}
```

Configure Policy Options

Configure a policy statement at the [edit policy-options] hierarchy level to import routes from the customer's CE router as follows:

```
[edit]
policy-options {
  policy-statement policy-name {
    term term-name {
      from {
        protocol bgp;
        community community-name;
      }
      then accept;
    }
    term term-name {
      then reject;
    }
  }
}
```

Configure a policy statement to export routes to the customer's CE router as follows:

```
[edit]
policy-options {
  policy-statement policy-name {
    term term-name {
      from protocol bgp;
      then {
        community add community-name;
        accept;
      }
    }
    term term-name {
      then reject;
    }
  }
  community community-name members value;
}
```

Carrier-of-Carriers VPN—Customer Provides VPN Service

Figure 40, “Carrier-of-Carriers VPN Architecture,” on page 266 shows how the routers in this type of service interconnect.

Configure the following routers in the customer's and provider's networks to enable carrier-of-carriers VPN service:

- Configure the Carrier-of-Carriers Customer's PE Router on page 279
- Configure the Carrier-of-Carriers Customer's CE Router on page 281
- Configure the Provider's PE Router on page 283

Configure the Carrier-of-Carriers Customer's PE Router

The carrier-of-carriers customer's PE router is connected to the end customer's CE router.

Configure MPLS

Configure MPLS at the [edit protocols mpls] hierarchy level on the carrier-of-carriers customer's PE router as follows:

```
[edit]
protocols {
  mpls {
    interface interface-name;
    interface interface-name;
  }
}
```

Configure BGP

Configure the labeled-unicast statement at the [edit protocols bgp] hierarchy level on the IBGP session to the carrier-of-carriers customer's CE router (see "Configure the Carrier-of-Carriers Customer's CE Router" on page 281) and configure the family-inet-vpn statement for the IBGP session to the carrier-of-carriers PE router on the other side of the network:

```
[edit]
protocols {
  bgp {
    group group-name {
      type internal;
      local-address address;
      neighbor address {
        family inet {
          labeled-unicast;
          resolve-vpn;
        }
      }
    }
    neighbor address {
      family inet-vpn {
        any;
      }
    }
  }
}
```

Configure OSPF

Configure OSPF at the [edit protocols ospf] hierarchy level on the carrier-of-carriers customer's PE router as follows:

```
[edit]
protocols {
  ospf {
    area area-id {
      interface interface-name {
        passive;
      }
      interface interface-name;
    }
  }
}
```

Configure LDP

Configure LDP at the [edit protocols ldp] hierarchy level on the carrier-of-carriers customer's PE router as follows:

```
[edit]
protocols {
  ldp {
    interface interface-name;
  }
}
```

Configure VPN Service in the Routing Instance

Configure VPN service for the end customer's CE router at the [edit routing-instances *routing-instance-name*] hierarchy level on the carrier-of-carriers customer's PE router:

```
[edit]
routing-instances {
  routing-instance-name {
    instance-type vrf;
    interface interface-name;
    route-distinguisher address;
    vrf-import policy-name;
    vrf-export policy-name;
    protocols {
      bgp {
        group group-name {
          peer-as as-number;
          neighbor address;
        }
      }
    }
  }
}
```


Configure Policy Options

Configure policy options at the [edit policy-options] hierarchy level to import and export routes to and from the end customer's CE router:

```
[edit]
policy-options {
  policy-statement policy-name {
    term term-name {
      from {
        protocol bgp;
        community community-name;
      }
      then accept;
    }
    term term-name {
      then reject;
    }
  }
  policy-statement policy-name {
    term term-name {
      from protocol bgp;
      then {
        community add community-name;
        accept;
      }
    }
    term term-name {
      then reject;
    }
  }
  community community-name members value;
}
```

Configure the Carrier-of-Carriers Customer's CE Router

The carrier-of-carriers customer's CE router connects to the provider's PE router.

Configure MPLS

In the MPLS configuration for the carrier-of-carriers customer's CE router, include the interfaces to the provider's PE router and to a provider router in the customer's network:

```
[edit]
protocols {
  mpls {
    traffic-engineering bgp-igp;
    interface interface-name;
    interface interface-name;
  }
}
```

Configure BGP

In the BGP configuration for the carrier-of-carriers customer's CE router at the [edit protocols bgp] hierarchy level, configure a group that includes the labeled-unicast statement to extend VPN service to the PE router connected to the end customer's CE outer:

```
[edit]
protocols {
  bgp {
    group group-name {
      type internal;
      local-address address;
      neighbor address {
        family inet {
          labeled-unicast;
        }
      }
    }
  }
}
```

Configure a group to send labeled internal routes to the provider's PE router as follows:

```
[edit]
protocols {
  bgp {
    group group-name {
      export internal;
      peer-as as-number;
      neighbor address {
        family inet {
          labeled-unicast;
        }
      }
    }
  }
}
```

Configure OSPF and LDP

Configure OSPF and LDP at the [edit protocols] hierarchy level on the carrier-of-carriers customer's CE router as follows:

```
[edit]
protocols {
  ospf {
    area area-id {
      interface interface-name {
        passive;
      }
      interface interface-name;
    }
  }
  ldp {
    interface interface-name;
  }
}
```

Configure Policy Options

Configure the policy options at the [edit policy-options] hierarchy level on the carrier-of-carriers customer's CE router as follows:

```
[edit]
policy-options {
  policy-statement policy-statement-name {
    term term-name {
      from protocol [ ospf direct ldp ];
      then accept;
    }
    term term-name {
      then reject;
    }
  }
}
```

Configure the Provider's PE Router

The carrier-of-carriers provider's PE routers connect to the carrier customer's CE routers.

Configure MPLS

Configure at least two interfaces at the [edit protocols mpls] hierarchy level—one to the customer's CE router and one to connect to the provider's PE router on the other side of the provider's network:

```
[edit]
protocols {
  mpls {
    interface interface-name;
    interface interface-name;
  }
}
```

Configure a PE-Router-to-PE-Router BGP Session

Configure a PE-router-to-PE-router BGP session at the [edit protocols bgp] hierarchy level on the provider's PE routers to allow VPN-IPv4 routes to pass between the PE routers:

```
[edit]
protocols {
  bgp {
    group group-name {
      type internal;
      local-address address;
      family inet-vpn {
        any;
      }
      neighbor address;
    }
  }
}
```

Configure IS-IS and LDP

Configure IS-IS and LDP at the [edit protocols] hierarchy level on the provider's PE routers as follows:

```
[edit]
protocols {
  isis {
    interface interface-name;
    interface interface-name {
      passive;
    }
  }
  ldp {
    interface interface-name;
  }
}
```

Configure Policy Options

Configure policy statements at the [edit policy-options] hierarchy level on the provider's PE router to export routes to and import routes from the carrier customer's network:

```
[edit]
policy-options {
  policy-statement statement-name {
    term term-name {
      from {
        protocol bgp;
        community community-name;
      }
      then accept;
    }
    term term-name {
      then reject;
    }
  }
  policy-statement statement-name {
    term term-name {
      from protocol bgp;
      then {
        community add community-name;
        accept;
      }
    }
    term term-name {
      then reject;
    }
  }
  community community-name members value;
}
```

Configure a Routing Instance to Send Labeled Routes to the CE Router

Configure the routing instance at the [edit routing-instances] hierarchy level on the provider's PE router to send labeled routes to the carrier customer's CE router:

```
[edit]
routing-instances {
  routing-instance-name {
    instance-type vrf;
    interface interface-name;
    route-distinguisher value;
    vrf-import policy-name;
    vrf-export policy-name;
    protocols {
      bgp {
        group group-name {
          peer-as as-number;
          neighbor address {
            family inet {
              labeled-unicast;
            }
          }
        }
      }
    }
  }
}
```


Chapter 14

Configuration Examples for Interprovider and Carrier-of-Carriers VPNs

This chapter contains examples that illustrate how to configure interprovider and carrier-of-carriers virtual private networks (VPNs). It includes the following sections:

- Example Terminology on page 287
- Interprovider VPN Examples on page 288
- Carrier-of-Carriers VPN Examples on page 298
- Multiple Instances for LDP and Carrier-of-Carriers VPNs on page 320

Example Terminology

The following terminology is used in these examples and is specific to Juniper Networks:

- **bgp.l3vpn.0**: The table on the provider edge (PE) router in which the VPN-IPv4 routes that are received from another PE router are stored. Incoming routes are checked against the `vrf-import` statements from all the VPNs configured on the PE router. If there is a match, the VPN-Internet Protocol version 4 (IPv4) route is added to the `bgp.l3vpn.0` table. To view the `bgp.l3vpn.0` table, issue the `show route table bgp.l3vpn.0` command.
- ***routing-instance-name*.inet.0**: The routing table for a specific routing instance. For example, a routing instance called VPN-A has a routing table called VPN-A.inet.0. Routes are added to this table in the following ways:
 - They are sent from a customer edge (CE) router configured within the VPN-A routing instance.
 - They are advertised from a remote PE router that passes the `vrf-import` policy configured within VPN-A (to view the route, run the `show route` command). IPv4 (not VPN-IPv4) routes are stored in this table.
- **vrf-import *policy-name***: An import policy configured on a particular routing instance on a PE router. This policy is required for the configuration of interprovider and carrier-of-carriers VPNs. It is applied to VPN-IPv4 routes learned from another PE router or a route reflector.

- `vrf-export policy-name`: An export policy configured on a particular routing instance on a PE router. It is required for the configuration of interprovider and carrier-of-carriers VPNs. It is applied to VPN-IPv4 routes, (originally learned from locally connected CE routers as IPv4 routes), which are advertised to another PE router or route reflector.
- MP-EBGP: The multiprotocol external Border Gateway Protocol (MP-EBGP) mechanism is used to export VPN-IPv4 routes across an autonomous system (AS) boundary. To apply this mechanism, use the `labeled-unicast` statement at the `[edit protocols bgp group group-name family inet]` hierarchy level.

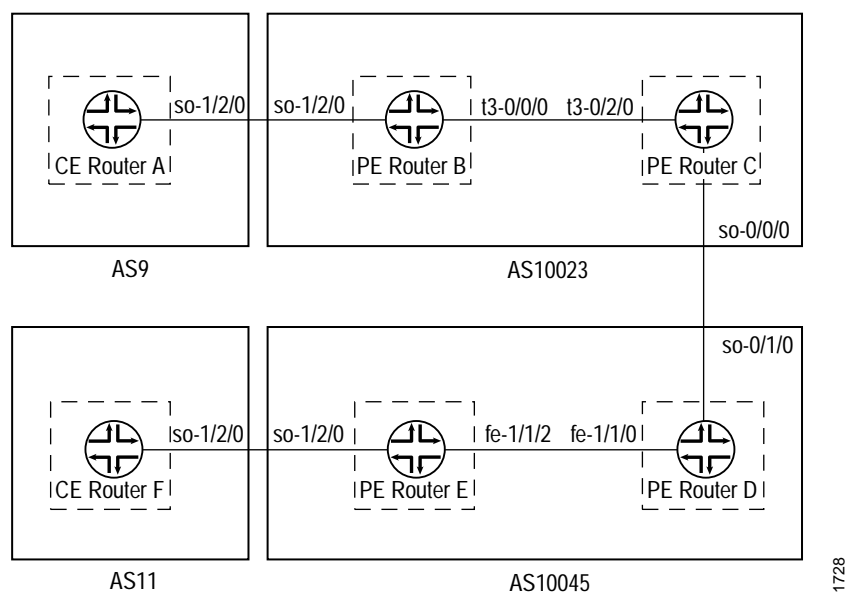
Interprovider VPN Examples

The following examples illustrate how to configure interprovider VPNs:

- Interprovider VPN Example—MP-EBGP between ISP Peer Routers on page 289
- Interprovider VPN Example—Multihop MP-EBGP on page 295

Figure 41 illustrates the network topology used in both VPN examples.

Figure 41: Network Topology of Interprovider VPN Examples



Interprovider VPN Example—MP-EBGP between ISP Peer Routers

In this example, all routes learned from the CE routers are sent over both service provider networks as VPN-IPv4 routes. The routes are initially learned by the PE routers (Router B and Router E) from the CE routers (Router A and Router F) and are announced by the PE routers to the AS border routers (Router C and Router D). The AS border routers are then configured with a multiprotocol EBGp session enabling them to pass the VPN-IPv4 routes with each other. When an AS border router—Router C for example—learns VPN-IPv4 routes from an internal Border Gateway Protocol (IBGP) PE, the following occurs:

1. Router C sets itself as the next-hop for the route and creates a label for that route.
2. Router C advertises the VPN-IPv4 route to PE Router D in AS 10045.
3. Router D sets the next-hop to itself, creates another label, and then forwards the label and the route to its IBGP PE router (Router E).

This example has scaling limitations because of restrictions on the number of labels each PE router needs to allocate at the AS border.

Configuration for Router A

Configure a family inet EBGp session with Router B and export the direct routes:

```
[edit]
protocols {
  bgp {
    group to-provider {
      export attached;
      peer-as 10023;
      neighbor 192.168.198.2;
    }
  }
}
policy-options {
  policy-statement attached {
    from protocol direct;
    then accept;
  }
}
```

Configuration for Router B

Router A is configured as a CE router (using the routing-instances statement) in the configuration for Router B. Because they exchange VPN-IPv4 routes, Router D and Router C are configured as PE routers.

Configure Router B as follows:

```
[edit]
protocols {
  rsvp {
    interface t3-0/0/0.0;
  }
  mpls {
    label-switched-path to-routerC {
      to 10.255.14.171;
      description "to-routerC for use with VPNs";
    }
    interface t3-0/0/0.0;
    interface so-1/2/0.0;
  }
  bgp {
    group to-ibgp {
      type internal;
      local-address 10.255.14.175;
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.14.171;
    }
  }
  ospf {
    traffic-engineering;
    reference-bandwidth 4g;
    area 0.0.0.0 {
      interface t3-0/0/0.0;
      interface lo0.0 {
        passive;
      }
    }
  }
}
routing-instances {
  vpna {
    instance-type vrf;
    interface so-1/2/0.0;
    route-distinguisher 10.255.14.175:9;
    vrf-import vpna-import;
    vrf-export vpna-export;
    protocols {
      bgp {
        group to-ce {
          peer-as 9;
          neighbor 192.168.198.1;
        }
      }
    }
  }
}
```

```

policy-options {
  policy-statement vpn-import {
    term 1 {
      from {
        protocol bgp;
        community vpn-comm;
      }
      then accept;
    }
    term 2 {
      then reject;
    }
  }
  policy-statement vpn-export {
    term 1 {
      from protocol bgp;
      then {
        community add vpn-comm;
        accept;
      }
    }
    term 2 {
      then reject;
    }
  }
  community vpn-comm members target:100:1001;
}

```

Configuration for Router C

In the BGP protocol configuration for Router C, include the keep all statement. This forces BGP to store every route learned through BGP. Configure two BGP sessions (configure family inet-vpn on both sessions):

- IBGP session to Router D (group to-ibgp in this example)
- EBGP session to Router B (group to-ebgp-pe in this example)

Also note that interface t3-0/2/0 is added at the [edit protocols mpls] hierarchy level, which allows BGP to announce routes with labels over the EBGP session.

Configure Router C as follows:

```

[edit]
protocols {
  rsvp {
    interface t3-0/2/0.0;
  }
  mpls {
    label-switched-path to-routerB {
      to 10.255.14.175;
      description "to-routerB for use with vpns";
    }
    interface t3-0/2/0.0;
    interface so-0/0/0.0;
  }
}

```

.....

.....

.....

• • • • •

```

bgp {
  keep all;
  group to-ibgp-pe {
    type internal;
    family inet-vpn {
      unicast;
    }
    neighbor 10.255.14.177;
  }
  group to-ebgp-pe {
    type external;
    family inet-vpn {
      unicast;
    }
    peer-as 10023;
    neighbor 192.168.197.21;
  }
}
ospf {
  traffic-engineering;
  reference-bandwidth 4g;
  area 0.0.0.0 {
    interface fe-1/1/0.0;
    interface lo0.0 {
      passive;
    }
  }
}
}

```

Configuration for Router E

The configuration for Router E is very similar to the configuration for Router B:

```

[edit]
protocols {
  rsvp {
    interface fe-1/1/2.0;
  }
  mpls {
    label-switched-path to-routerD {
      to 10.255.14.173;
      description "to-routerD for use with VPNa";
    }
    interface fe-1/1/2.0;
    interface so-1/2/0.0;
  }
  bgp {
    group to-ibgp-pe {
      type internal;
      local-address 10.255.14.177;
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.14.173;
    }
  }
}

```

```

ospf {
  traffic-engineering;
  reference-bandwidth 4g;
  area 0.0.0.0 {
    interface fe-1/1/2.0;
    interface lo0.0 {
      passive;
    }
  }
}
}
routing-instances {
  vpn {
    instance-type vrf;
    interface so-1/2/0.0;
    route-distinguisher 10.255.14.177:11;
    vrf-import vpn-import;
    vrf-export vpn-export;
    protocols {
      bgp {
        group to-routerF-ce {
          neighbor 192.168.198.14 {
            peer-as 11;
          }
        }
      }
    }
  }
}
}
policy-options {
  policy-statement vpn-import {
    term 1 {
      from {
        protocol bgp;
        community vpn-comm;
      }
      then accept;
    }
    term 2 {
      then reject;
    }
  }
  policy-statement vpn-export {
    term 1 {
      from protocol bgp;
      then {
        community add vpn-comm;
        accept;
      }
    }
    term 2 {
      then reject;
    }
  }
  community vpn-comm members target:100:1001;
}

```

Configuration for Router F

Configure Router F as a CE router; the configuration is similar to that for Router A:

```
[edit]
protocols {
  bgp {
    group to-provider {
      type external;
      export attached;
      neighbor 192.168.198.13 {
        peer-as 10045;
      }
    }
  }
}
policy-options {
  policy-statement attached {
    from protocol direct;
    then accept;
  }
}
```

Interprovider VPN Example—Multihop MP-EBGP

In this example, labeled IPv4 (not VPN-IPv4), routes are exchanged by the AS border routers (Router C and Router D) to provide Multiprotocol Label Switching (MPLS) connectivity between the PE routers. Only routes internal to the service provider networks should be announced between Router C and Router D. Configure this by including the family inet labeled-unicast statement in the IBGP and EBGp configuration on the PE routers. When you set family inet labeled-unicast, the local router announces internal routes from inet.0 in the following manner:

- If a label exists for the route, the local router creates a label, performs a swap, and announces the route from inet.0 with the label.
- If a label does not exist for the route, the local router creates a label, performs a pop, and announces the route from inet.0 with the label.

Routes learned from the labeled-unicast session are placed into the inet.0 routing table.

In addition, you configure a multihop MP-EBGP session between the end PE routers (Router B and Router E). This additional MP-EBGP session allows the announcement of VPN-IPv4 routes, and allows you to maintain VPN connectivity while keeping VPN-IPv4 routes out of the core of the network.



Note

The configurations for the routers in the “Interprovider VPN Example—Multihop MP-EBGP” section are similar to those for the routers in the “Interprovider VPN Example—MP-EBGP between ISP Peer Routers” section. In the sections that follow, only the differences in these configurations are shown. The configurations for Router A and Router F are the same so they are not repeated.

Configuration for Router B

In group to-ibgp, include the family inet labeled-unicast statement to pass labeled IPv4 routes and configure an EBGp multihop session to pass VPN-IPv4 routes:

```
[edit]
protocols {
  bgp {
    group to-ibgp {
      type internal;
      local-address 10.255.14.175;
      family inet {
        labeled-unicast {
          resolve-vpn;
        }
      }
      neighbor 10.255.14.171;
    }
    group to-remote-pe {
      multihop {
        ttl 10;
      }
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.14.177 {
        peer-as 10045;
      }
    }
  }
}
```

Configuration for Router C

Configure Router C as follows:

```
[edit]
protocols {
  bgp {
    group to-ibgp {
      type internal;
      local-address 10.255.14.171;
      family inet {
        labeled-unicast;
      }
      neighbor 10.255.14.175;
    }
    group to-ebgp-pe {
      type external;
      family inet {
        labeled-unicast;
      }
      export internal;
      neighbor 192.168.197.22 {
        peer-as 10045;
      }
    }
  }
}
```



```

policy-options {
  policy-statement internal {
    term 1 {
      from protocol [ ospf direct ];
      then accept;
    }
    term 2 {
      then reject;
    }
  }
}

```

Configuration for Router D

Configure Router D as follows:

```

[edit]
protocols {
  bgp {
    group to-ibgp-pe {
      type internal;
      family inet {
        labeled-unicast;
      }
      neighbor 10.255.14.177;
    }
    group to-ebgp-pe {
      type external;
      family inet {
        labeled-unicast;
      }
      export internal;
      peer-as 10023;
      neighbor 192.168.197.21;
    }
  }
}
policy-options {
  policy-statement internal {
    term 1 {
      from protocol [ direct ospf ];
      then accept;
    }
    term 2 {
      then reject;
    }
  }
}

```

Configuration for Router E

Configure Router E as follows:

```
[edit]
protocols {
  bgp {
    group to-ibgp-pe {
      type internal;
      local-address 10.255.14.177;
      family inet {
        labeled-unicast;
      }
      neighbor 10.255.14.173;
    }
    group to-remote-pe {
      multihop {
        ttl 10;
      }
      family inet-vpn {
        unicast;
      }
      neighbor 10.255.14.175 {
        peer-as 10023;
      }
    }
  }
}
```

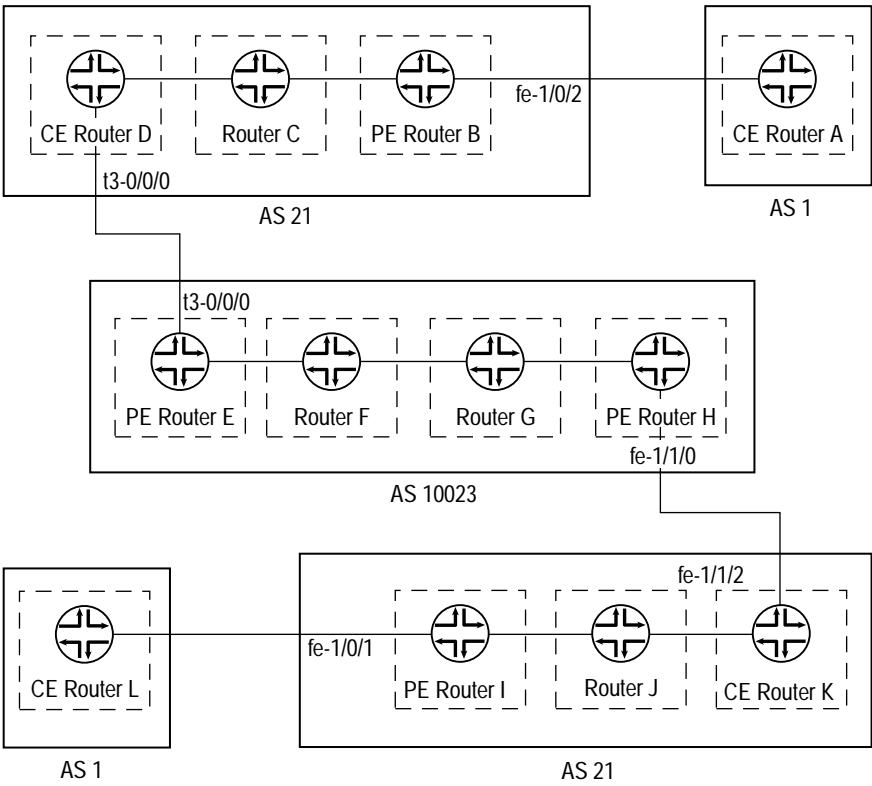
Carrier-of-Carriers VPN Examples

A carrier-of-carriers service allows an Internet service provider (ISP) to connect to a transparent outsourced backbone at multiple locations. There are two variations of this example:

- Carrier-of-Carriers VPN Example—Customer Provides Internet Service on page 300
- Carrier-of-Carriers VPN Example—Customer Provides VPN Service on page 309

Figure 42 shows the network topology in both carrier-of-carriers examples.

Figure 42: Carrier-of-Carriers VPN Example Network Topology



1729

Carrier-of-Carriers VPN Example—Customer Provides Internet Service

In this example, the carrier customer is not required to configure MPLS and Label Distribution Protocol (LDP) on its network. However, the carrier provider must configure MPLS and LDP on its network.

Configuration for Router A

In this example, Router A represents an end customer. You configure this router as a CE device.

```
[edit]
protocols {
  bgp {
    group to-routerB {
      export attached;
      peer-as 21;
      neighbor 192.168.197.169;
    }
  }
}
policy-options {
  policy-statement attached {
    from protocol direct;
    then accept;
  }
}
```

Configuration for Router B

Router B can act as the gateway router, responsible for aggregating end customers and connecting them to the network. Note that if a full-mesh IBGP session is configured, you can use route reflectors.

```
[edit]
protocols {
  bgp {
    group int {
      type internal;
      local-address 10.255.14.179;
      neighbor 10.255.14.175;
      neighbor 10.255.14.181;
      neighbor 10.255.14.176;
      neighbor 10.255.14.178;
      neighbor 10.255.14.177;
    }
    group to-vpn-blue {
      peer-as 1;
      neighbor 192.168.197.170;
    }
  }
}
```

```

ospf {
  area 0.0.0.0 {
    interface lo0.0 {
      passive;
    }
    interface fe-1/0/3.0;
    interface fe-1/0/2.0 {
      passive;
    }
  }
}

```

Configuration for Router C

Configure Router C as follows:

```

[edit]
protocols {
  bgp {
    group int {
      type internal;
      local-address 10.255.14.176;
      neighbor 10.255.14.179;
      neighbor 10.255.14.175;
      neighbor 10.255.14.177;
      neighbor 10.255.14.178;
      neighbor 10.255.14.181;
    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface fe-0/3/3.0;
      interface fe-0/3/0.0;
    }
  }
}

```

Configuration for Router D

Router D is the CE router with respect to AS 10023. In a carrier-of-carriers VPN, the CE router must be able to send labels to the carrier provider; this is done with the labeled-unicast statement in group to-isp-red.

```
[edit]
protocols {
  mpls {
    interface t3-0/0/0.0;
  }
  bgp {
    group int {
      type internal;
      local-address 10.255.14.175;
      neighbor 10.255.14.179;
      neighbor 10.255.14.176;
      neighbor 10.255.14.177;
      neighbor 10.255.14.178;
      neighbor 10.255.14.181;
    }
    group to-isp-red {
      export internal;
      peer-as 10023;
      neighbor 192.168.197.13 {
        family inet {
          labeled-unicast;
        }
      }
    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface fe-0/3/0.0;
      interface t3-0/0/0.0 {
        passive;
      }
    }
  }
}
policy options {
  policy-statement internal {
    term a {
      from protocol [ ospf direct ];
      then accept;
    }
    term b {
      then reject;
    }
  }
}
```

Configuration for Router E

This configuration sets up the inet-vpn IGBP session with Router H and the PE router portion of the VPN with Router D. Because Router D is required to send labels in this example, configure the BGP session with the labeled-unicast statement within the VPN routing and forwarding (VRF) table.

```
[edit]
protocols {
  mpls {
    interface t3-0/2/0.0;
    interface at-0/1/0.0;
  }
  bgp {
    group pe-pe {
      type internal;
      local-address 10.255.14.171;
      family inet-vpn {
        any;
      }
      neighbor 10.255.14.173;
    }
  }
  isis {
    interface at-0/1/0.0;
    interface lo0.0 {
      passive;
    }
  }
  ldp {
    interface at-0/1/0.0;
  }
}
routing-instances {
  vpn-isp1 {
    instance-type vrf;
    interface t3-0/2/0.0;
    route-distinguisher 10.255.14.171:21;
    vrf-import vpn-isp1-import;
    vrf-export vpn-isp1-export;
    protocols {
      bgp {
        group to-isp1 {
          peer-as 21;
          neighbor 192.168.197.14 {
            family inet {
              labeled-unicast;
            }
          }
        }
      }
    }
  }
}
```

```

policy-options {
  policy-statement vpn-isp1-import {
    term a {
      from {
        protocol bgp;
        community vpn-isp1-comm;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement vpn-isp1-export {
    term a {
      from protocol bgp;
      then {
        community add vpn-isp1-comm;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community vpn-isp1-comm members target:69:21;
}

```

Configuration for Router F

Configure Router F to act as a label-swapping router as follows:

```

[edit]
protocols {
  isis {
    interface so-0/2/0.0;
    interface at-0/3/0.0;
    interface lo0.0 {
      passive;
    }
  }
  ldp {
    interface so-0/2/0.0;
    interface at-0/3/0.0;
  }
}

```


Configuration for Router G

Configure Router G to act as a label-swapping router as follows:

```
[edit]
protocols {
  isis {
    interface so-0/0/0.0;
    interface so-1/0/0.0;
    interface lo0.0 {
      passive;
    }
  }
  ldp {
    interface so-0/0/0.0;
    interface so-1/0/0.0;
  }
}
```

Configuration for Router H

Router H acts as the PE router for AS 10023. The configuration that follows is similar to that for Router F:

```
[edit]
protocols {
  mpls {
    interface fe-1/1/0.0;
    interface so-1/0/0.0;
  }
  bgp {
    group pe-pe {
      type internal;
      local-address 10.255.14.173;
      family inet-vpn {
        any;
      }
      neighbor 10.255.14.171;
    }
  }
  isis {
    interface so-1/0/0.0;
    interface lo0.0 {
      passive;
    }
  }
  ldp {
    interface so-1/0/0.0;
  }
}
```

```

routing-instances {
  vpn-isp1 {
    instance-type vrf;
    interface fe-1/1/0.0;
    route-distinguisher 10.255.14.173:21;
    vrf-import vpn-isp1-import;
    vrf-export vpn-isp1-export;
    protocols {
      bgp {
        group to-isp1 {
          peer-as 21;
          neighbor 192.168.197.94 {
            family inet {
              labeled-unicast;
            }
          }
        }
      }
    }
  }
}

policy-options {
  policy-statement vpn-isp1-import {
    term a {
      from {
        protocol bgp;
        community vpn-isp1-comm;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement vpn-isp1-export {
    term a {
      from protocol bgp;
      then {
        community add vpn-isp1-comm;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community vpn-isp1-comm members target:69:21;
}

```

Configuration for Router I

Configure Router I to connect to the basic Internet service customer (Router L) as follows:

```
[edit]
protocols {
  mpls {
    interface fe-1/0/1.0;
    interface fe-1/1/3.0;
  }
  bgp {
    group int {
      type internal;
      local-address 10.255.14.181;
      neighbor 10.255.14.177;
      neighbor 10.255.14.179;
      neighbor 10.255.14.175;
      neighbor 10.255.14.176;
      neighbor 10.255.14.178;
    }
    group to-vpn-green {
      peer-as 1;
      neighbor 192.168.197.198;
    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface fe-1/0/1.0 {
        passive;
      }
      interface fe-1/1/3.0;
    }
  }
}
```

Configuration for Router J

Configure Router J as a label-swapping router as follows:

```
[edit]
protocols {
  bgp {
    group int {
      type internal;
      local-address 10.255.14.178;
      neighbor 10.255.14.177;
      neighbor 10.255.14.181;
      neighbor 10.255.14.175;
      neighbor 10.255.14.176;
      neighbor 10.255.14.179;
    }
  }
}
```

```

ospf {
  area 0.0.0.0 {
    interface lo0.0 {
      passive;
    }
    interface fe-1/0/2.0;
    interface fe-1/0/3.0;
  }
}

```

Configuration for Router K

Router K acts as the CE router at the end of the connection to the carrier provider. As in the configuration for Router D, you include the labeled-unicast statement for the EBGp session:

```

[edit]
protocols {
  mpls {
    interface fe-1/1/2.0;
    interface fe-1/0/2.0;
  }
  bgp {
    group int {
      type internal;
      local-address 10.255.14.177;
      neighbor 10.255.14.181;
      neighbor 10.255.14.178;
      neighbor 10.255.14.175;
      neighbor 10.255.14.176;
      neighbor 10.255.14.179;
    }
    group to-isp-red {
      export internal;
      peer-as 10023;
      neighbor 192.168.197.93 {
        family inet {
          labeled-unicast;
        }
      }
    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface fe-1/0/2.0;
      interface fe-1/1/2.0 {
        passive;
      }
    }
  }
}

```

```

policy-options {
  policy-statement internal {
    term a {
      from protocol [ ospf direct ];
      then accept;
    }
    term b {
      then reject;
    }
  }
}

```

Configuration for Router L

Configure Router L to act as the end customer for the carrier-of-carriers VPN service as follows:

```

[edit]
protocols {
  bgp {
    group to-routerl {
      export attached;
      peer-as 21;
      neighbor 192.168.197.197;
    }
  }
}
policy-options {
  policy-statement attached {
    from protocol direct;
    then accept;
  }
}

```

Carrier-of-Carriers VPN Example—Customer Provides VPN Service

In this example, the carrier customer *must* run some form of MPLS (Resource Reservation Protocol [RSVP] or LDP) on its network to provide VPN services to the end customer. In example below, Router B and Router I act as PE routers, and a functioning MPLS path is required between these routers if they exchange VPN-IPv4 routes.

Configuration for Router A

In this example, Router A acts as the CE router for the end customer. Configure a default family inet BGP session on Router A:

```
[edit]
protocols {
  bgp {
    group to-routerB {
      export attached;
      peer-as 21;
      neighbor 192.168.197.169;
    }
  }
}
policy-options {
  policy-statement attached {
    from protocol direct;
    then accept;
  }
}
```

Configuration for Router B

Router B is the PE router for the end customer CE router (Router A), so you need to configure a routing instance (vpna). Configure the labeled-unicast statement on the IBGP session to Router D, and configure family-inet-vpn for the IBGP session to the other side of the network (see Figure 42 on page 299) with Router I:

```
[edit]
protocols {
  mpls {
    interface fe-1/0/2.0;
    interface fe-1/0/3.0;
  }
  bgp {
    group int {
      type internal;
      local-address 10.255.14.179;
      neighbor 10.255.14.175 {
        family inet {
          labeled-unicast;
          resolve-vpn;
        }
      }
    }
    neighbor 10.255.14.181 {
      family inet-vpn {
        any;
      }
    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface fe-1/0/3.0;
    }
  }
}
```

```

ldp {
    interface fe-1/0/3.0;
}
routing-instances {
    vpn {
        instance-type vrf;
        interface fe-1/0/2.0;
        route-distinguisher 10.255.14.179:21;
        vrf-import vpn-import;
        vrf-export vpn-export;
        protocols {
            bgp {
                group vpn-06 {
                    peer-as 1;
                    neighbor 192.168.197.170;
                }
            }
        }
    }
}
policy-options {
    policy-statement vpn-import {
        term a {
            from {
                protocol bgp;
                community vpn-comm;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement vpn-export {
        term a {
            from protocol bgp;
            then {
                community add vpn-comm;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
}
community vpn-comm members target:100:1001;
}

```

Configuration for Router C

Configure Router C as a label-swapping router within the local AS as follows:

```
[edit]
protocols {
  mpls {
    traffic-engineering bgp-igp;
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface fe-0/3/3.0;
      interface fe-0/3/0.0;
    }
  }
  ldp {
    interface fe-0/3/0.0;
    interface fe-0/3/3.0;
  }
}
```

Configuration for Router D

Router D acts as the CE router for the VPN services provided by the AS 10023 network. In group int, you configure the labeled-unicast statement to Router B (10.255.14.179). You also need to configure the BGP group to-isp-red to send labeled internal routes to the PE router (Router E).

```
[edit]
protocols {
  mpls {
    traffic-engineering bgp-igp;
    interface fe-0/3/0.0;
    interface t3-0/0/0.0;
  }
  bgp {
    group int {
      type internal;
      local-address 10.255.14.175;
      neighbor 10.255.14.179 {
        family inet {
          labeled-unicast;
        }
      }
    }
    group to-isp-red {
      export internal;
      peer-as 10023;
      neighbor 192.168.197.13 {
        family inet {
          labeled-unicast;
        }
      }
    }
  }
}
```



```

ospf {
  area 0.0.0.0 {
    interface lo0.0 {
      passive;
    }
    interface fe-0/3/0.0;
  }
}
ldp {
  interface fe-0/3/0.0;
}
}
policy-options {
  policy-statement internal {
    term a {
      from protocol [ ospf direct ];
      then accept;
    }
    term b {
      then reject;
    }
  }
}
}

```

Configuration for Router E

Router E and Router H are PE routers. Configure a PE-router-to-PE-router BGP session to allow VPN-IPv4 routes to pass between these two PE routers. Configure the routing instance on Router E to send labeled routes to the CE router (Router D).

Configure Router E as follows:

```

[edit]
protocols {
  mpls {
    interface t3-0/2/0.0;
    interface at-0/1/0.0;
  }
  bgp {
    group pe-pe {
      type internal;
      local-address 10.255.14.171;
      family inet-vpn {
        any;
      }
      neighbor 10.255.14.173;
    }
  }
  isis {
    interface at-0/1/0.0;
    interface lo0.0 {
      passive;
    }
  }
  ldp {
    interface at-0/1/0.0;
  }
}

```

```

policy-options {
  policy-statement vpn-isp1-import {
    term a {
      from {
        protocol bgp;
        community vpn-isp1-comm;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement vpn-isp1-export {
    term a {
      from protocol bgp;
      then {
        community add vpn-isp1-comm;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community vpn-isp1-comm members target:69:21;
}
routing-instances {
  vpn-isp1 {
    instance-type vrf;
    interface t3-0/2/0.0;
    route-distinguisher 10.255.14.171:21;
    vrf-import vpn-isp1-import;
    vrf-export vpn-isp1-export;
    protocols {
      bgp {
        group to-isp1 {
          peer-as 21;
          neighbor 192.168.197.14 {
            family inet {
              labeled-unicast;
            }
          }
        }
      }
    }
  }
}

```

Configuration for Router F

Configure Router F to swap labels for routes running through its interfaces as follows:

```
[edit]
protocols {
  isis {
    interface so-0/2/0.0;
    interface at-0/3/0.0;
    interface lo0.0 {
      passive;
    }
  }
  ldp {
    interface so-0/2/0.0;
    interface at-0/3/0.0;
  }
}
```

Configuration for Router G

Configure Router G as follows:

```
[edit]
protocols {
  isis {
    interface so-0/0/0.0;
    interface so-1/0/0.0;
    interface lo0.0 {
      passive;
    }
  }
  ldp {
    interface so-0/0/0.0;
    interface so-1/0/0.0;
  }
}
```

Configuration for Router H

The configuration for Router H is similar to the configuration for Router E:

```
[edit]
protocols {
  mpls {
    interface fe-1/1/0.0;
    interface so-1/0/0.0;
  }
  bgp {
    group pe-pe {
      type internal;
      local-address 10.255.14.173;
      family inet-vpn {
        any;
      }
      neighbor 10.255.14.171;
    }
  }
}
```

```

isis {
    interface so-1/0/0.0;
    interface lo0.0 {
        passive;
    }
}
ldp {
    interface so-1/0/0.0;
}
}
routing-instances {
    vpn-isp1 {
        instance-type vrf;
        interface fe-1/1/0.0;
        route-distinguisher 10.255.14.173:21;
        vrf-import vpn-isp1-import;
        vrf-export vpn-isp1-export;
        protocols {
            bgp {
                group to-isp1 {
                    peer-as 21;
                    neighbor 192.168.197.94 {
                        family inet {
                            labeled-unicast;
                        }
                    }
                }
            }
        }
    }
}
}
policy-options {
    policy-statement vpn-isp1-import {
        term a {
            from {
                protocol bgp;
                community vpn-isp1-comm;
            }
            then accept;
        }
        term b {
            then reject;
        }
    }
    policy-statement vpn-isp1-export {
        term a {
            from protocol bgp;
            then {
                community add vpn-isp1-comm;
                accept;
            }
        }
        term b {
            then reject;
        }
    }
}
community vpn-isp1-comm members target:69:21;
}

```

Configuration for Router I

Router I acts as the PE router for the end customer. The configuration that follows is similar to the configuration for Router B:

```
[edit]
protocols {
  mpls {
    interface fe-1/0/1.0;
    interface fe-1/1/3.0;
  }
  bgp {
    group int {
      type internal;
      local-address 10.255.14.181;
      neighbor 10.255.14.177 {
        family inet {
          labeled-unicast {
            resolve-vpn;
          }
        }
      }
      neighbor 10.255.14.179 {
        family inet-vpn {
          any;
        }
      }
    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface fe-1/1/3.0;
    }
  }
  ldp {
    interface fe-1/1/3.0;
  }
}
routing-instances {
  vpna {
    instance-type vrf;
    interface fe-1/0/1.0;
    route-distinguisher 10.255.14.181:21;
    vrf-import vpna-import;
    vrf-export vpna-export;
    protocols {
      bgp {
        group vpna-0 {
          peer-as 1;
          neighbor 192.168.197.198;
        }
      }
    }
  }
}
```

```

policy-options {
  policy-statement vpn-import {
    term a {
      from {
        protocol bgp;
        community vpn-comm;
      }
      then accept;
    }
    term b {
      then reject;
    }
  }
  policy-statement vpn-export {
    term a {
      from protocol bgp;
      then {
        community add vpn-comm;
        accept;
      }
    }
    term b {
      then reject;
    }
  }
  community vpn-comm members target:100:1001;
}

```

Configuration for Router J

Configure Router J to swap labels for routes running through its interfaces as follows:

```

[edit]
protocols {
  mpls {
    traffic-engineering bgp-igp;
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface fe-1/0/2.0;
      interface fe-1/0/3.0;
    }
  }
  ldp {
    interface fe-1/0/2.0;
    interface fe-1/0/3.0;
  }
}

```

Configuration for Router K

The configuration for Router K is similar to the configuration for Router D:

```
[edit]
protocols {
  mpls {
    traffic-engineering bgp-igp;
    interface fe-1/1/2.0;
    interface fe-1/0/2.0;
  }
  bgp {
    group int {
      type internal;
      local-address 10.255.14.177;
      neighbor 10.255.14.181 {
        family inet {
          labeled-unicast;
        }
      }
    }
    group to-isp-red {
      export internal;
      peer-as 10023;
      neighbor 192.168.197.93 {
        family inet {
          labeled-unicast;
        }
      }
    }
  }
  ospf {
    area 0.0.0.0 {
      interface lo0.0 {
        passive;
      }
      interface fe-1/0/2.0;
    }
  }
  ldp {
    interface fe-1/0/2.0;
  }
}
policy-options {
  policy-statement internal {
    term a {
      from protocol [ ospf direct ];
      then accept;
    }
    term b {
      then reject;
    }
  }
}
```

Configuration for Router L

In this example, Router L is the end customer's CE router. Configure Router L as follows:

```
[edit]
protocols {
  bgp {
    group to-l {
      export attached;
      peer-as 21;
      neighbor 192.168.197.197;
    }
  }
}
policy-options {
  policy-statement attached {
    from protocol direct;
    then accept;
  }
}
```

Multiple Instances for LDP and Carrier-of-Carriers VPNs

By configuring multiple LDP routing instances, you can use LDP to advertise labels in a carrier-of-carriers VPN from a core provider PE router to a customer carrier CE router. This is especially useful when the carrier customer is a basic ISP and wants to restrict full Internet routes to its PE routers. By using LDP instead of BGP, the carrier customer shields its other internal routers from the Internet at large. Multiple-instance LDP is also useful when a carrier customer wants to provide Layer 3 VPN or Layer 2 VPN services to its customers.

For an example of how to configure multiple LDP routing instances for carrier-of-carriers VPNs, see the *JUNOS Internet Software Feature Guide* on the product documentation page of the Juniper Networks Web site, located at <http://www.juniper.net/>.

Chapter 15

Summary of the Interprovider and Carrier-of-Carriers Configuration Statement

The following section explains the configuration statement that applies specifically to hierarchical and recursive Border Gateway Protocol (BGP) and Multiprotocol Label Switching (MPLS) virtual private networks (VPNs).

labeled-unicast

Syntax	labeled-unicast { resolve-vpn; }
Hierarchy Level	[edit protocols bgp group <i>group-name</i> family inet]
Description	This statement advertises labeled routes from the inet.0 VPN and places labeled routes into the inet.0 VPN. When the labeled-unicast statement is used, the local router automatically performs a next-hop to self on all routes advertised into the external BGP (EBGP) from the internal BGP (IBGP) and from IBGP to EBGP.
Options	resolve-vpn—(Optional) Stores labeled routes in the inet.3 table to resolve routes for a provider edge (PE) router located in a different autonomous system (AS). Note that for a PE router to install a route in the VPN routing and forwarding table (VRF), the next-hop must resolve to a route stored within the inet.3 table.
Usage Guidelines	See “Interprovider and Carrier-of-Carriers Configuration Guidelines” on page 269.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

Part 5

Layer 2 Circuits

- Layer 2 Circuit Configuration on page 327
- Summary of Layer 2 Circuit Configuration Statements on page 331

.....

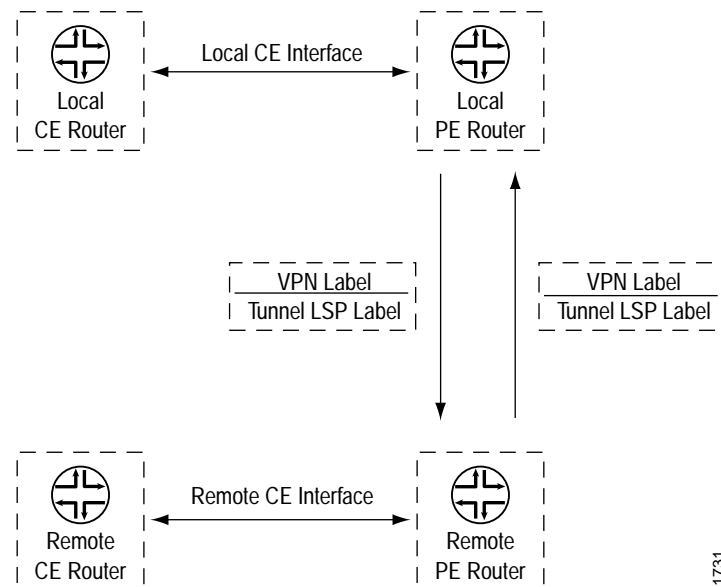
Chapter 16

Layer 2 Virtual Circuit Overview

A Layer 2 circuit is a point-to-point Layer 2 connection transported by means of Multiprotocol Label Switching (MPLS) or another tunneling technology on the service provider's network. A Layer 2 circuit is similar to a circuit cross-connect (CCC) except that multiple Layer 2 circuits can be transported over a single label-switched path (LSP) tunnel between two provider edge (PE) routers. In contrast, each CCC requires a dedicated LSP.

The JUNOS software implementation of Layer 2 virtual circuits supports only the remote form of a Layer 2 circuit; that is, a connection from a local customer edge (CE) router to a remote CE router. Figure 43 illustrates the components of a Layer 2 virtual circuit.

Figure 43: Components of a Layer 2 Virtual Circuit



The interfaces shown in Figure 43 are logical interfaces. Packets are sent to the remote CE router using an egress virtual private network (VPN) label advertised by the remote PE router. The VPN label transits over either a Resource Reservation Protocol (RSVP) or a Label Distribution Protocol (LDP) LSP (or other type) tunnel to the remote PE router connected to the remote CE router. Return traffic sent from the remote CE router to the local CE router uses an ingress VPN label advertised by the local PE router, which again transits over an RSVP and LDP LSP to the local PE router from the remote PE router. LDP is the signaling protocol used for advertising VPN labels.

For more information on Layer 2 circuits, see *Transport of Layer 2 Frames Over MPLS*, Internet draft draft-martini-l2circuit-trans-mpls-07.txt. This draft is available on the IETF web site at <http://www.ietf.org/>.

Chapter 17

Layer 2 Circuit Configuration

To configure a Layer 2 circuit, include statements at the [edit protocols l2circuit] hierarchy level:

```
[edit]
protocols {
  l2circuit {
    neighbor address {
      interface interface-name {
        (control-word | no-control-word);
        virtual-circuit-id identifier;
      }
    }
    traceoptions {
      file filename <replace> <size size> <files number> <nostamp>;
      flag flag <flag-modifier> <disable>;
    }
  }
}
```

The following sections describe how to configure Layer 2 virtual circuits:

- Configure the Neighbor and Interface on page 327
- Configure the Virtual Circuit ID on page 328
- Configure the Interface Encapsulation Type on page 328
- Configure LDP for Layer 2 Circuits on page 328
- Disable the Control Word for Layer 2 Circuits on page 329
- Trace Layer 2 Circuit Creation and Changes on page 329

Configure the Neighbor and Interface

Each Layer 2 circuit is represented by the logical interface connecting the local PE router to the local CE router. All the Layer 2 circuits using a particular remote PE router designated for remote CE routers are listed under the *neighbor* statement (*neighbor* designates the PE router). Each neighbor is identified by its IP address and is usually the end-point destination for the LSP tunnel transporting the Layer 2 circuit.

Configure the Virtual Circuit ID

You configure a virtual circuit ID on each interface. Each virtual circuit ID uniquely identifies the Layer 2 circuit among all the Layer 2 circuits to a specific neighbor. The key to identifying a particular Layer 2 circuit on a PE router is the neighbor address and the virtual circuit ID. An LDP-FEC-to-label binding is associated with a Layer 2 circuit based on the virtual circuit ID in the forwarding equivalence class (FEC) and the neighbor that sent this binding. It enables the dissemination of the VPN label used for sending traffic on that Layer 2 circuit to the remote CE router.

Configure the virtual circuit ID at the [edit protocols l2circuit neighbor *address* interface *interface-name*] hierarchy level:

```
[edit protocols l2circuit neighbor address interface interface-name]
virtual-circuit-id identifier;
```

Configure the Interface Encapsulation Type

Both ends of a Layer 2 circuit must connect using the same Layer 2 encapsulation. The Layer 2 encapsulation type is carried in the LDP FEC. The encapsulation type received from an FEC is matched against the local encapsulation type of the Layer 2 circuit. The Layer 2 circuit will not work if the encapsulation types do not match.

The configuration for the encapsulation type on Layer 2 virtual circuits is identical to the configuration for the CCC encapsulation type. For more information, see the *JUNOS Internet Software Configuration Guide: MPLS Applications*.

To configure the interface encapsulation for a Layer 2 circuit, include statements at the [edit interfaces] hierarchy level:

```
[edit]
interfaces {
  interface-name {
    encapsulation encapsulation-type;
    unit unit-number;
  }
}
```

Configure LDP for Layer 2 Circuits

Use LDP as the signaling protocol to advertise ingress labels to the remote PE routers. When configured, LDP examines the Layer 2 circuit configuration and initiates extended neighbor discovery for all the Layer 2 circuit neighbors (for example, remote PEs). This is similar to how LDP works when tunneled over RSVP. You must run LDP on the lo0.0 interface for extended neighbor discovery to function correctly.

For detailed information about how to configure LDP, see the *JUNOS Internet Software Configuration Guide: MPLS Applications*.

Disable the Control Word for Layer 2 Circuits

The emulated VC encapsulation for Layer 2 circuits is accomplished by adding a 4-byte control word between the Layer 2 protocol data unit (PDU) being transported and the VC label that is used for demultiplexing. Various networking formats (ATM, Frame Relay, Ethernet, and so on) use the control word in a variety of ways.

JUNOS software does not support the control word for any networking format, meaning that it is not fully compliant with the Internet draft in cases where the control word is mandatory. To be minimally compliant with the Internet draft, JUNOS supports a null control word (a control word of all zeros). If JUNOS receives a packet with a control word attached, the control word is discarded before the packet is forwarded to its destination.

JUNOS can typically determine whether a neighboring router supports the control word or not. However, if you want to explicitly disable its use on a specific interface, include the `no-control-word` statement at the `[edit protocols l2circuit neighbor address interface interface-name]` hierarchy level:

```
[edit protocols l2circuit neighbor address interface interface-name]
no-control-word;
```

This statement might be required for Layer 2 VPN configurations. For more information, see “Disable the Control Word for Layer 2 VPNs” on page 22.

Trace Layer 2 Circuit Creation and Changes

To trace the creation of and changes to Layer 2 virtual circuits, you can specify options in the `traceoptions` statement at the `[edit protocols l2circuit]` hierarchy level:

```
[edit protocols l2circuit]
traceoptions {
  file filename <replace> <size size> <files number> <nostamp> <no-world-readable>
  <world-readable>;
  flag flag <flag-modifier> <disable>;
}
```

The following tracing flags display the operations associated with Layer 2 virtual circuits:

- `connections`—Layer 2 circuit connections (events and state changes)
- `error`—Error conditions
- `FEC`—Layer 2 circuit advertisements received or sent using LDP
- `topology`—Layer 2 circuit topology changes caused by reconfiguration or advertisements received from other PE routers

.....

Chapter 18

Summary of Layer 2 Circuit Configuration Statements

The following sections explain the major protocol configuration statements that apply specifically to Layer 2 circuits. The statements are organized alphabetically. Protocols and the statements at the [edit protocols] hierarchy level are explained in the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

control-word

Syntax	(control-word no-control-word);
Hierarchy Level	[edit protocols l2circuit neighbor <i>address</i> interface <i>interface-name</i>] [edit routing-instances <i>routing-instance-name</i> protocols l2vpn]
Description	<p>The control word is 4 bytes long and is inserted between the Layer 2 protocol data unit (PDU) being transported and the VC label that is used for demultiplexing.</p> <ul style="list-style-type: none">■ control-word—Enables the use of the control word. Default: The control word is enabled by default. You can also configure the control word explicitly using the control-word statement.■ no-control-word—Disables the use of the control word.
Usage Guidelines	See “Disable the Control Word for Layer 2 Circuits” on page 329 or “Disable the Control Word for Layer 2 VPNs” on page 22.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

interface

Syntax	interface <i>interface-name</i> ;
Hierarchy Level	[edit protocols l2circuit neighbor <i>address</i>]
Description	Interface over which Layer 2 circuit traffic travels.
Options	<i>interface-name</i> —Name of the interface to configure.
Usage Guidelines	See “Configure the Neighbor and Interface” on page 327.
Required Privilege Level	routing—To view this statement in the configuration. routing-control—To add this statement to the configuration.

neighbor

Syntax neighbor *address*

Hierarchy Level [edit protocols l2circuit]

Description Each Layer 2 circuit is represented by the logical interface connecting the local provider edge (PE) router to the local customer edge (CE) router. All the Layer 2 circuits using a particular remote PE router designated for remote CE routers are listed under the neighbor statement (neighbor designates the PE router). Each neighbor is identified by its IP address and is usually the end-point destination for the label-switched path (LSP) tunnel (transporting the Layer 2 circuit).

Options *address*—IP address of a neighboring router.

Usage Guidelines See “Configure the Neighbor and Interface” on page 327.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

no-control-word

See “control-word” on page 331.

traceoptions

Syntax traceoptions {
 file *filename* <replace> <size *size*> <files *number*> <nostamp> <no-world-readable |
 world-readable>;
 flag *flag* <*flag-modifier*> <disable>;
}

Hierarchy Level [edit protocols l2circuit]

Description Trace traffic flowing through a Layer 2 virtual circuit.

Options disable—(Optional) Disable the tracing operation. You can use this option to disable a single operation when you have defined a broad group of tracing operations, such as all.

file *filename*—Name of the file to receive the output of the tracing operation. Enclose the name within quotation marks.

files *number*—(Optional) Maximum number of trace files. When a trace file named *trace-file* reaches its maximum size, it is renamed *trace-file.0*, then *trace-file.1*, and so on, until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum number of files, you also must specify a maximum file size with the size option.

Range: 2 to 1000

Default: 2 files

flag *flag*—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements.

- **connections**—Layer 2 circuit connections (events and state changes)
- **error**—Error conditions
- **FEC**—Layer 2 circuit advertisements received or sent by means of the Label Distribution Protocol (LDP)
- **topology**—Layer 2 circuit topology changes caused by reconfiguration or advertisements received from other PE routers

flag-modifier—(Optional) Modifier for the tracing flag. You can specify the detail modifier if you want to provide detailed trace information.

nostamp—(Optional) Do not place timestamp information at the beginning of each line in the trace file.

Default: If you omit this option, timestamp information is placed at the beginning of each line of the tracing output.

no-world-readable—(Optional) Do not allow any user to read the log file.

replace—(Optional) Replace an existing trace file if there is one.

Default: If you do not include this option, tracing output is appended to an existing trace file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named *trace-file* reaches this size, it is renamed *trace-file.0*. When the *trace-file* again reaches its maximum size, *trace-file.0* is renamed *trace-file.1* and *trace-file* is renamed *trace-file.0*. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you also must specify a maximum number of trace files with the **files** option.

Syntax: *xk* to specify KB, *xm* to specify MB, or *xg* to specify GB

Range: 10 KB through the maximum file size supported on your system

Default: 1 MB

world-readable—(Optional) Allow any user to read the log file.

Usage Guidelines See “Trace Layer 2 Circuit Creation and Changes” on page 329.

Required Privilege Level **routing**—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

virtual-circuit-id

Syntax virtual-circuit-id *identifier*;

Hierarchy Level [edit protocols l2circuit neighbor *address* interface *interface-name*]

Description Uniquely identifies a Layer 2 virtual circuit.

Options None.

Usage Guidelines See “Configure the Virtual Circuit ID” on page 328.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Part 6

Appendix

■ Glossary on page 337

.....

Appendix A

Glossary

A

AAL	ATM adaptation layer. A series of protocols enabling various types of traffic, including voice, data, image, and video, to run over an ATM network.
active route	Route chosen from all routes in the routing table to reach a destination. Active routes are installed into the forwarding table.
add/drop multiplexer	<i>See ADM.</i>
Address Resolution Protocol	<i>See ARP.</i>
adjacency	Portion of the local routing information that pertains to the reachability of a single neighbor over a single circuit or interface.
ADM	Add/drop multiplexer. SONET functionality that allows lower-level signals to be dropped from a high-speed optical connection.
aggregation	Combination of groups of routes that have common addresses into a single entry in the routing table.
AH	Authentication Header. A component of the IPSec protocol used to verify that the contents of a packet have not been changed, and to validate the identity of the sender. <i>See also ESP.</i>
ANSI	American National Standards Institute. The United States' representative to the ISO.
APQ	Alternate Priority Queuing. Dequeuing method that has a special queue, similar to SPQ, which is visited only 50 percent of the time. The packets in the special queue still have a predictable latency, although the upper limit of the delay is higher than that with SPQ. Since the other configured queues share the remaining 50 percent of the service time, queue starvation is usually avoided. <i>See also SPQ.</i>
APS	Automatic Protection Switching. Technology used by SONET ADMs to protect against circuit faults between the ADM and a router and to protect against failing routers.
area	Routing subdomain that maintains detailed routing information about its own internal composition and that maintains routing information that allows it to reach other routing subdomains. In IS-IS, an area corresponds to a Level 1 subdomain. In IS-IS and OSPF, a set of contiguous networks and hosts within an autonomous system that have been administratively grouped together.
area border router	Router that belongs to more than one area. Used in OSPF.

ARP	Address Resolution Protocol. Protocol for mapping IP addresses to MAC addresses.
AS	Autonomous system. Set of routers under a single technical administration. Each AS normally uses a single interior gateway protocol (IGP) and metrics to propagate routing information within the set of routers. Also called <i>routing domain</i> .
AS boundary router	In OSPF, routers that exchange routing information with routers in other ASs.
AS external link advertisements	OSPF link-state advertisement sent by AS boundary routers to describe external routes that they know. These link-state advertisements are flooded throughout the AS (except for stub areas).
AS path	In BGP, the route to a destination. The path consists of the AS numbers of all routers a packet must go through to reach a destination.
ASIC	Application-specific integrated circuit. Specialized processors that perform specific functions on the router.
ATM	Asynchronous Transfer Mode. A high-speed multiplexing and switching method utilizing fixed-length cells of 53 octets to support multiple types of traffic.
atomic	Smallest possible operation. An atomic operation is performed either entirely or not at all. For example, if machine failure prevents a transaction from completing, the system is rolled back to the start of the transaction, with no changes taking place.
Authentication Header	<i>See AH.</i>
Automatic Protection Switching	<i>See APS.</i>
autonomous system	<i>See AS.</i>
autonomous system boundary router	In OSPF, routers that exchange routing information with routers in other ASs.
autonomous system external link advertisements	OSPF link-state advertisement sent by autonomous system boundary routers to describe external routes that they know. These link-state advertisements are flooded throughout the autonomous system (except for stub areas).
autonomous system path	In BGP, the route to a destination. The path consists of the autonomous system numbers of all the routers a packet must pass through to reach a destination.
B	
backbone area	In OSPF, an area that consists of all networks in area ID 0.0.0.0, their attached routers, and all area border routers.
backplane	On an M40 router, component of the Packet Forwarding Engine that distributes power, provides signal connectivity, manages shared memory on FPCs, and passes outgoing data cells to FPCs.
bandwidth	The range of transmission frequencies a network can use, expressed as the difference between the highest and lowest frequencies of a transmission channel. In computer networks, greater bandwidth indicates faster data-transfer rate capacity.
Bellcore	Bell Communications Research. Research and development organization created after the divestiture of the Bell System. It is supported by the regional Bell holding companies (RBHCs), which own the regional Bell operating companies (RBOCs).

BERT	Bit error rate test. A test that can be run on a T3 interface to determine whether it is operating properly.
BGP	Border Gateway Protocol. Exterior gateway protocol used to exchange routing information among routers in different autonomous systems.
bit error rate test	<i>See BERT.</i>
BITS	Building Integrated Timing Source. Dedicated timing source that synchronizes all equipment in a particular building.
Border Gateway Protocol	<i>See BGP.</i>
broadcast	Operation of sending network traffic from one network node to all other network nodes.
bundle	Collection of software that makes up a JUNOS software release.
C	
CB	Control Board. Part of the host subsystem that provides control and monitoring functions for router components.
CCC	Circuit cross-connect. A JUNOS software feature that allows you to configure transparent connections between two circuits, where a circuit can be a Frame Relay DLCI, an ATM VC, a PPP interface, a Cisco HDLC interface, or an MPLS label-switched path (LSP).
CE device	Customer edge device. Router or switch in the customer's network that is connected to a service provider's provider edge (PE) router and participates in a Layer 3 VPN.
CFM	Cubic feet per minute. Measure of air flow in volume per minute.
Challenge Handshake Authentication Protocol	<i>See CHAP.</i>
channel service unit	<i>See CSU/DSU.</i>
CHAP	A protocol that authenticates remote users. CHAP is a server-driven, three-step authentication mechanism that depends on a shared secret password that resides on both the server and the client.
CIDR	Classless interdomain routing. A method of specifying Internet addresses in which you explicitly specify the bits of the address to represent the network address instead of determining this information from the first octet of the address.
CIP	Connector Interface Panel. On an M160 router, the panel that contains connectors for the Routing Engines, BITS interfaces, and alarm relay contacts.
circuit cross-connect	<i>See CCC.</i>
class of service	<i>See CoS.</i>
CLEC	(Pronounced "see-lek") Competitive Local Exchange Carrier. Company that competes with the already established local telecommunications business by providing its own network and switching.
CLEI	Common language equipment identifier. Inventory code used to identify and track telecommunications equipment.

CLI	Command-line interface. Interface provided for configuring and monitoring the routing protocol software.
client peer	In a BGP route reflection, a member of a cluster that is not the route reflector. <i>See also nonclient peer.</i>
CLNP	Connectionless Network Protocol. ISO-developed protocol for OSI connectionless network service. CLNP is the OSI equivalent of IP.
cluster	In BGP, a set of routers that have been grouped together. A cluster consists of one system that acts as a route reflector, along with any number of client peers. The client peers receive their route information only from the route reflector system. Routers in a cluster do not need to be fully meshed.
community	In BGP, a group of destinations that share a common property. Community information is included as one of the path attributes in BGP update messages.
confederation	In BGP, a group of systems that appears to external autonomous systems to be a single autonomous system.
constrained path	In traffic engineering, a path determined using RSVP signaling and constrained using CSPF. The ERO carried in the packets contains the constrained path information.
core	The central backbone of the network.
CoS	Class of service. The method of classifying traffic on a packet-by-packet basis using information in the ToS byte to provide different service levels to different traffic.
CPE	Customer premises equipment. Telephone or other service provider equipment located at a customer site.
craft interface	Mechanisms used by a Communication Workers of America craftsman to operate, administer, and maintain equipment or provision data communications. On a Juniper Networks router, the craft interface allows you to view status and troubleshooting information and perform system control functions.
CSCP	Class Selector Codepoint.
CSNP	Complete sequence number PDU. Packet that contains a complete list of all the LSPs in the IS-IS database.
CSPF	Constrained Shortest Path First. An MPLS algorithm that has been modified to take into account specific restrictions when calculating the shortest path across the network.
CSU/DSU	Channel service unit/data service unit. Channel service unit connects a digital phone line to a multiplexer or other digital signal device. Data service unit connects a DTE to a digital phone line.
customer edge device	<i>See CE device.</i>

D

daemon	Background process that performs operations on behalf of the system software and hardware. Daemons normally start when the system software is booted, and they run as long as the software is running. In the JUNOS software, daemons are also referred to as processes.
damping	Method of reducing the number of update messages sent between BGP peers, thereby reducing the load on these peers without adversely affecting the route convergence time for stable routes.
data circuit-terminating equipment	<i>See DCE.</i>
data-link connection identifier	<i>See DLCI.</i>
data service unit	<i>See CSU/DSU.</i>
Data Terminal Equipment	<i>See DTE.</i>
dcd	The JUNOS software interface process (daemon).
DCE	Data circuit-terminating equipment. RS-232-C device, typically used for a modem or printer, or a network access and packet switching node.
default address	Router address that is used as the source address on unnumbered interfaces.
denial of service	<i>See DoS.</i>
dense wavelength-division multiplexing	<i>See DWDM.</i>
designated router	In OSPF, a router selected by other routers that is responsible for sending link-state advertisements that describe the network, which reduces the amount of network traffic and the size of the routers' topological databases.
destination prefix length	Number of bits of the network address used for host portion of a CIDR IP address.
DHCP	Dynamic Host Configuration Protocol. Allocates IP addresses dynamically so that they can be reused when they are no longer needed.
Diffie-Hellman	A public key scheme, invented by Whitfield Diffie and Martin Hellman, used for sharing a secret key without communicating secret information, thus precluding the need for a secure channel. Once correspondents have computed the secret shared key, they can use it to encrypt communications.
Diffserv	Differentiated Service (based on RFC 2474). Diffserv uses the ToS byte to identify different packet flows on a packet-by-packet basis. Diffserv adds a Class Selector Codepoint (CSCP) and a Differentiated Services Codepoint (DSCP).
Dijkstra algorithm	<i>See SPF.</i>
DIMM	Dual inline memory module. 168-pin memory module that supports 64-bit data transfer.
direct routes	<i>See interface routes.</i>

	DLCI	Data-link connection identifier. Identifier for a Frame Relay virtual connection (also called a logical interface).
	DoS	Denial of service. System security breach in which network services become unavailable to users.
	DRAM	Dynamic random-access memory. Storage source on the router that can be accessed quickly by a process.
	drop profile	Drop probabilities for different levels of buffer fullness that are used by RED to determine from which queue to drop packets.
	DSCP	Differentiated Services Codepoint.
	DSU	Data service unit. A device used to connect a DTE to a digital phone line. Converts digital data from a router to voltages and encoding required by the phone line. <i>See also CSU/DSU.</i>
	DTE	Data Terminal Equipment. RS-232-C interface that a computer uses to exchange information with a serial device.
	DVMRP	Distance Vector Multicast Routing Protocol. Distributed multicast routing protocol that dynamically generates IP multicast delivery trees using a technique called reverse path multicasting (RPM) to forward multicast traffic to downstream interfaces.
	DWDM	Dense wavelength-division multiplexing. Technology that enables data from different sources to be carried together on an optical fiber, with each signal carried on its own separate wavelength.
	Dynamic Host Configuration Protocol	<i>See DHCP.</i>
E	EBGP	External BGP. BGP configuration in which sessions are established between routers in different ASs.
	ECSA	Exchange Carriers Standards Association. A standards organization created after the divestiture of the Bell System to represent the interests of interexchange carriers.
	edge router	In MPLS, a router located at the beginning or end of a label-switching tunnel. When at the beginning of a tunnel, an edge router applies labels to new packets entering the tunnel. When at the end of a tunnel, the edge router removes labels from packets exiting the tunnel. <i>See also MPLS.</i>
	EGP	Exterior gateway protocol, such as BGP.
	egress router	In MPLS, last router in a label-switched path (LSP). <i>See also ingress router.</i>
	EIA	Electronic Industries Association. A United States trade group that represents manufacturers of electronics devices and sets standards and specifications.
	EMI	Electromagnetic interference. Any electromagnetic disturbance that interrupts, obstructs, or otherwise degrades or limits the effective performance of electronics or electrical equipment.
	encapsulating security payload	<i>See ESP.</i>
	end system	In IS-IS, network entity that sends and receives packets.

ERO	Explicit Route Object. Extension to RSVP that allows an RSVP PATH message to traverse an explicit sequence of routers that is independent of conventional shortest-path IP routing.
ESP	Encapsulating security payload. A fundamental component of IPSec-compliant VPNs, ESP specifies an IP packet's encryption, data integrity checks, and sender authentication, which are added as a header to the IP packet. <i>See also AH.</i>
explicit path	<i>See signaled path.</i>
Explicit Route Object	<i>See ERO.</i>
export	To place routes from the routing table into a routing protocol.
external BGP	<i>See EBGP.</i>
external metric	A cost included in a route when OSPF exports route information from external autonomous systems. There are two types of external metrics: Type 1 and Type 2. Type 1 external metrics are equivalent to the link-state metric; that is, the cost of the route, used in the internal autonomous system. Type 2 external metrics are greater than the cost of any path internal to the autonomous system.
F	
fast reroute	Mechanism for automatically rerouting traffic on an LSP if a node or link in an LSP fails, thus reducing the loss of packets traveling over the LSP.
FEAC	Far-end alarm and control. T3 signal used to send alarm or status information from the far-end terminal back to the near-end terminal and to initiate T3 loopbacks at the far-end terminal from the near-end terminal.
FEB	Forwarding Engine Board. In M5 and M10 routers, provides route lookup, filtering, and switching to the destination port.
firewall	A security gateway positioned between two different networks, usually between a trusted network and the Internet. A firewall ensures that all traffic that crosses it conforms to the organization's security policy. Firewalls track and control communications, deciding whether to pass, reject, discard, encrypt, or log them. Firewalls also can be used to secure sensitive portions of a local network.
FIFO	First in, first out.
flap damping	<i>See damping.</i>
flapping	<i>See route flapping.</i>
Flexible PIC Concentrator	<i>See FPC.</i>
Forwarding Engine Board	<i>See FEB.</i>
forwarding information base	<i>See forwarding table.</i>
forwarding table	JUNOS software forwarding information base (FIB). The JUNOS routing protocol process installs active routes from its routing tables into the Routing Engine forwarding table. The kernel copies this forwarding table into the Packet Forwarding Engine, which is responsible for determining which interface transmits the packets.

FPC Flexible PIC Concentrator. An interface concentrator on which PICs are mounted. An FPC inserts into a slot in a Juniper Networks router. *See also PIC.*

FRU Field-replaceable unit. Router component that customers can replace onsite.

G

group A collection of related BGP peers.

H

hash A one-way function that takes a message of any length and produces a fixed-length digest. In security, a message digest is used to validate that the contents of a message have not been altered in transit. The Secure Hash Algorithm (SHA-1) and Message Digest 5 (MD5) are commonly used hashes.

Hashed Message Authentication Code *See HMAC.*

HDLCL High-level data link control. An International Telecommunication Union (ITU) standard for a bit-oriented data link layer protocol on which most other bit-oriented protocols are based.

HMAC Hashed Message Authentication Code. A mechanism for message authentication that uses cryptographic hash functions. HMAC can be used with any iterative cryptographic hash function—for example, MD5 or SHA-1—in combination with a secret shared key. The cryptographic strength of HMAC depends on the properties of the underlying hash function.

hold time Maximum number of seconds allowed to elapse between the time a BGP system receives successive keepalive or update messages from a peer.

host module On an M160 router, provides routing and system management functions of the router. Consists of the Routing Engine and Miscellaneous Control Subsystem (MCS).

host subsystem Provides routing and system-management functions of the router. Consists of a Routing Engine and an adjacent Control Board (CB).

I

IANA Internet Assigned Numbers Authority. Regulatory group that maintains all assigned and registered Internet numbers, such as IP and multicast addresses. *See also NIC.*

IBGP Internal BGP. BGP configuration in which sessions are established between routers in the same ASs.

ICMP Internet Control Message Protocol. Used in router discovery, ICMP allows router advertisements that enable a host to discover addresses of operating routers on the subnet.

IDE Integrated Drive Electronics. Type of hard disk on the Routing Engine.

IEC International Electrotechnical Commission. *See ISO.*

IEEE Institute of Electronic and Electrical Engineers. International professional society for electrical engineers.

IETF Internet Engineering Task Force. International community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet.

IGMP	Internet Group Membership Protocol. Used with multicast protocols to determine whether group members are present.
IGP	Interior gateway protocol, such as IS-IS, OSPF, and RIP.
IKE	Internet Key Exchange. The key management protocol used in IPsec, IKE combines the ISAKMP and Oakley protocols to create encryption keys and security associations.
import	To install routes from the routing protocols into a routing table.
ingress router	In MPLS, first router in a label-switched path (LSP). <i>See also egress router.</i>
inter-AS routing	Routing of packets among different ASs. <i>See also EBGp.</i>
intercluster reflection	In a BGP route reflection, the redistribution of routing information by a route reflector system to all nonclient peers (BGP peers not in the cluster). <i>See also route reflection.</i>
interface routes	Routes that are in the routing table because an interface has been configured with an IP address. Also called <i>direct routes</i> .
intermediate system	In IS-IS, network entity that sends and receives packets and that can also route packets.
internal BGP	<i>See IBGP.</i>
Internet Key Exchange	<i>See IKE.</i>
Internet Protocol Security	<i>See IPsec.</i>
Internet Security Association and Key Management Protocol	<i>See ISAKMP.</i>
intra-AS routing	The routing of packets within a single AS. <i>See also IBGP.</i>
IP	Internet Protocol. The protocol used for sending data from one point to another on the Internet.
IPsec	Internet Protocol Security. The industry standard for establishing VPNs, IPsec comprises a group of protocols and algorithms that provide authentication and encryption of data across IP-based networks.
ISAKMP	Internet Security Association and Key Management Protocol. A protocol that allows the receiver of a message to obtain a public key and use digital certificates to authenticate the sender's identity. ISAKMP is designed to be key exchange independent; that is, it supports many different key exchanges. <i>See also IKE and Oakley.</i>
IS-IS	Intermediate System-to-Intermediate System protocol. Link-state, interior gateway routing protocol for IP networks that also uses the shortest-path first (SPF) algorithm to determine routes.
ISO	International Organization for Standardization. Worldwide federation of standards bodies that promotes international standardization and publishes international agreements as International Standards.

ISP Internet service provider. Company that provides access to the Internet and related services.

ITU International Telecommunications Union (formerly known as the CCITT). Group supported by the United Nations that makes recommendations and coordinates the development of telecommunications standards for the entire world.

J

jitter Small random variation introduced into the value of a timer to prevent multiple timer expirations from becoming synchronized.

K

kernel forwarding table *See forwarding table.*

L

label In MPLS, 20-bit unsigned integer in the range 0 through 1048575, used to identify a packet traveling along an LSP.

label-switched path (LSP) Sequence of routers that cooperatively perform MPLS operations for a packet stream. The first router in an LSP is called the *ingress router*; and the last router in the path is called the *egress router*. An LSP is a point-to-point, half-duplex connection from the ingress router to the egress router. (The ingress and egress routers cannot be the same router.)

label switching *See MPLS.*

label-switching router *See LSR.*

link Communication path between two neighbors. A link is *up* when communication is possible between the two end points.

link-state PDU (LSP) Packets that contain information about the state of adjacencies to neighboring systems.

local preference Optional BGP path attribute carried in internal BGP update packets that indicates the degree of preference for an external route.

loose In the context of traffic engineering, a path that can use any route or any number of other intermediate (transit) points to reach the next address in the path. (Definition from RFC 791, modified to fit LSPs.)

LSP *See label-switched path (LSP) or link-state PDU (LSP).*

LSR Label-switching router. A router on which MPLS and RSVP are enabled and is thus capable of processing label-switched packets.

M

martian address Network address about which all information is ignored.

mask *See subnet mask.*

MBGP Multiprotocol BGP. An extension to BGP that allows you to connect multicast topologies within and between BGP ASs.

MBone Internet multicast backbone. An interconnected set of subnetworks and routers that support the delivery of IP multicast traffic. The MBone is a virtual network that is layered on top of sections of the physical Internet.

MCS	Miscellaneous Control Subsystem. On an M160 router, provides control and monitoring functions for router components and SONET clocking for the router.
MD5	Message Digest 5. A one-way hashing algorithm that produces a 128-bit hash. It is used in AH and ESP. <i>See also</i> <i>SHA-1</i> .
MDRR	Modified Deficit Round Robin. A method for selecting queues to be serviced.
MED	Multiple exit discriminator. Optional BGP path attribute consisting of a metric value that is used to determine the exit point to a destination when all other factors in determining the exit point are equal.
mesh	Network topology in which devices are organized in a manageable, segmented manner with many, often redundant, interconnections between network nodes.
Message Digest 5	<i>See</i> <i>MD5</i> .
MIB	Management Information Base. Definition of an object that can be managed by SNMP.
midplane	Forms the rear of the PIC cage on M5 and M10 routers and the FPC card cage on M20 and M160 routers. Provides data transfer, power distribution, and signal connectivity.
Miscellaneous Control Subsystem	<i>See</i> <i>MCS</i> .
MPLS	Multiprotocol Label Switching. Mechanism for engineering network traffic patterns that functions by assigning to network packets short labels that describe how to forward them through the network. Also called <i>label switching</i> . <i>See also</i> <i>traffic engineering</i> .
MTBF	Mean time between failure. Measure of hardware component reliability.
MTU	Maximum transfer unit. Limit on segment size for a network.
multicast	Operation of sending network traffic from one network node to multiple network nodes.
multicast distribution tree	The data path between the sender (host) and the multicast group member (receiver or listener).
multiprotocol BGP	<i>See</i> <i>MBGP</i> .
Multiprotocol Label Switching	<i>See</i> <i>MPLS</i> .
N	
neighbor	Adjacent system reachable by traversing a single subnetwork. An immediately adjacent router. Also called a <i>peer</i> .
NET	Network entity title. Network address defined by the ISO network architecture and used in CLNS-based networks.
network layer reachability information	<i>See</i> <i>NLRI</i> .
network link advertisement	An OSPF link-state advertisement flooded throughout a single area by designated routers to describe all routers attached to the network.

Network Time Protocol *See NTP.*

NIC Network Information Center. Internet authority responsible for assigning Internet-related numbers, such as IP addresses and autonomous system numbers. *See also IANA.*

NLRI Network layer reachability information. Information that is carried in BGP packets and is used by MBGP.

nonclient peer In a BGP route reflection, a BGP peer that is not a member of a cluster. *See also client peer.*

not-so-stubby area *See NSSA.*

NSAP Network service access point. Connection to a network that is identified by a network address.

n-selector Last byte of an nonclient peer address.

NSSA Not-so-stubby area. In OSPF, a type of stub area in which external routes can be flooded.

NTP Network Time Protocol. Protocol used to synchronize computer clock times on a network.

O

Oakley A key determination protocol based on the Diffie-Hellman algorithm that provides added security, including authentication. Oakley was the key-exchange algorithm mandated for use with the initial version of ISAKMP, although various algorithms can be used. Oakley describes a series of key exchanges called “modes” and details the services provided by each; for example, Perfect Forward Secrecy for keys, identity protection, and authentication. *See also ISAKMP.*

OC Optical Carrier. In SONET, Optical Carrier levels indicate the transmission rate of digital signals on optical fiber.

OSI Open System Interconnection. Standard reference model for how messages are transmitted between two points on a network.

OSPF Open Shortest Path First. A link-state IGP that makes routing decisions based on the shortest-path-first (SPF) algorithm (also referred to as the *Dijkstra algorithm*).

P

package A collection of files that make up a JUNOS software component.

Packet Forwarding Engine The architectural portion of the router that processes packets by forwarding them between input and output interfaces.

path attribute Information about a BGP route, such as the route origin, AS path, and next-hop router.

PCI Peripheral Component Interconnect. Standard, high-speed bus for connecting computer peripherals. Used on the Routing Engine.

PCMCIA Personal Computer Memory Card International Association. Industry group that promotes standards for credit card-size memory or I/O devices.

PDU Protocol data unit. IS-IS packets.

PE router Provider edge router. A router in the service provider's network that is connected to a customer edge (CE) device and that participates in a Virtual Private Network (VPN).

PEC	Policing Equivalence Classes. In traffic policing, a set of packets that is treated the same by the packet classifier.
peer	An immediately adjacent router with which a protocol relationship has been established. Also called a <i>neighbor</i> .
Perfect Forward Secrecy	<i>See PFS.</i>
PFE	<i>See Packet Forwarding Engine.</i>
PFS	A condition derived from an encryption system that changes encryption keys often and ensures that no two sets of keys have any relation to each other. The advantage of PFS is that if one set of keys is compromised, only communications using those keys are at risk. An example of a system that uses PFS is Diffie-Hellman.
Physical Interface Card	<i>See PIC.</i>
PIC	Physical Interface Card. A network interface-specific card that can be installed on an FPC in the router.
PIM	Protocol Independent Multicast. A protocol-independent multicast routing protocol. PIM Sparse Mode routes to multicast groups that might span wide-area and interdomain internets. PIM Dense Mode is a flood-and-prune protocol.
PLP	Packet Loss Priority.
PLP bit	Packet Loss Priority bit. Used to identify packets that have experienced congestion or are from a transmission that exceeded a service provider's customer service license agreement. This bit can be used as part of a router's congestion control mechanism and can be set by the interface or by a filter.
policing	Applying rate limits on bandwidth and burst size for traffic on a particular interface.
pop	Removal of the last label, by a router, from a packet as it exits an MPLS domain.
PPP	Point-to-Point Protocol. Link-layer protocol that provides multiprotocol encapsulation. It is used for link-layer and network-layer configuration.
precedence bits	The first three bits in the ToS byte. On a Juniper Networks router, these bits are used to sort or classify individual packets as they arrive at an interface. The classification determines the queue to which the packet is directed upon transmission.
preference	Desirability of a route to become the active route. A route with a lower preference value is more likely to become the active route. The preference is an arbitrary value in the range 0 through 255 that the routing protocol process uses to rank routes received from different protocols, interfaces, or remote systems.
preferred address	On an interface, the default local address used for packets sourced by the local router to destinations on the subnet.
primary address	On an interface, the address used by default as the local address for broadcast and multicast packets sourced locally and sent out the interface.
primary interface	Router interface that packets go out when no interface name is specified and when the destination address does not imply a particular outgoing interface.

Protocol-Independent Multicast *See PIM.*

provider edge router *See PE router.*

provider router Router in the service provider's network that does not attach to a customer edge (CE) device.

PSNP Partial sequence number PDU. Packet that contains only a partial list of the LSPs in the IS-IS link-state database.

push Addition of a label or stack of labels, by a router, to a packet as it enters an MPLS domain.

Q

QoS Quality of service. Performance, such as transmission rates and error rates, of a communications channel or system.

quality of service *See QoS.*

R

RADIUS Remote Authentication Dial-In User Service. Authentication method for validating users who attempt to access the router using Telnet.

Random Early Detection *See RED.*

rate limiting *See policing.*

RBOC (Pronounced "are-bock") Regional Bell operating company. Regional telephone companies formed as a result of the divestiture of the Bell System.

RDRAM RAMBUS dynamic random access memory.

RED Random Early Detection. Gradual drop profile for a given class that is used for congestion avoidance. RED tries to anticipate incipient congestion and reacts by dropping a small percentage of packets from the head of the queue to ensure that a queue never actually becomes congested.

Rendezvous Point *See RP.*

Resource Reservation Protocol *See RSVP.*

RFC Request for Comments. Internet standard specifications published by the Internet Engineering Task Force.

RFI Radio frequency interference. Interference from high-frequency electromagnetic waves emanating from electronic devices.

RIP Routing Information Protocol. Distance-vector interior gateway protocol that makes routing decisions based on hop count.

route flapping Situation in which BGP systems send an excessive number of update messages to advertise network reachability information.

route identifier IP address of the router from which a BGP, IGP, or OSPF packet originated.

route reflection	In BGP, configuring a group of routers into a cluster and having one system act as a route reflector, redistributing routes from outside the cluster to all routers in the cluster. Routers in a cluster do not need to be fully meshed.
router link advertisement	OSPF link-state advertisement flooded throughout a single area by all routers to describe the state and cost of the router's links to the area.
routing domain	<i>See AS.</i>
Routing Engine	Architectural portion of the router that handles all routing protocol processes, as well as other software processes that control the router's interfaces, some of the chassis components, system management, and user access to the router.
routing instance	A collection of routing tables, interfaces, and routing protocol parameters. The set of interfaces belongs to the routing tables and the routing protocol parameters control the information in the routing tables.
routing table	Common database of routes learned from one or more routing protocols. All routes are maintained by the JUNOS routing protocol process.
RP	For PIM-SM, a core router acting as the root of the distribution tree in a shared tree.
rpd	JUNOS software routing protocol process (daemon). User-level background process responsible for starting, managing, and stopping the routing protocols on a Juniper Networks router.
RPM	Reverse path multicasting. Routing algorithm used by DVMRP to forward multicast traffic.
RSVP	Resource Reservation Protocol. Resource reservation setup protocol designed to interact with integrated services on the Internet.

S

SA	Security association. An IPSec term that describes an agreement between two parties about what rules to use for authentication and encryption algorithms, key exchange mechanisms, and secure communications.
SAP	Session Announcement Protocol. Used with multicast protocols to handle session conference announcements.
SAR	Segmentation and reassembly. Buffering used with ATM.
SCB	System Control Board. On an M40 router, the part of the Packet Forwarding Engine that performs route lookups, monitors system components, and controls FPC resets.
SCG	SONET Clock Generator. Provides Stratum 3 clock signal for the SONET/SDH interfaces on the router. Also provides external clock inputs.
SDH	Synchronous Digital Hierarchy. CCITT variation of SONET standard.
SDP	Session Description Protocol. Used with multicast protocols to handle session conference announcements.
SDRAM	Synchronous dynamic random access memory.
Secure Hash Algorithm	<i>See SHA-1.</i>
secure shell	<i>See SSH.</i>

security association	<i>See SA.</i>
Security Parameter Index	<i>See SPI.</i>
SFM	Switching and Forwarding Module. On an M160 router, a component of the Packet Forwarding Engine that provides route lookup, filtering, and switching to FPCs.
SHA-1	Secure Hash Algorithm. A widely used hash function for use with Digital Signal Standard (DSS). SHA-1 is more secure than MD5.
shortest-path-first algorithm	<i>See SPF.</i>
signaled path	In traffic engineering, an explicit path; that is, a path determined using RSVP signaling. The ERO carried in the packets contains the explicit path information.
SIB	Switch Interface Board. Provides the switching function to the destination Packet Forwarding Engine.
simplex interface	An interface that assumes that packets it receives from itself are the result of a software loopback process. The interface does not consider these packets when determining whether the interface is functional.
SNMP	Simple Network Management Protocol. Protocol governing network management and the monitoring of network devices and their functions.
SONET	Synchronous Optical Network. High-speed (up to 2.5 Gbps) synchronous network specification developed by Bellcore and designed to run on optical fiber. STS-1 is the basic building block of SONET. Approved as an international standard in 1988. <i>See also SDH.</i>
SPF	Shortest-path first, an algorithm used by IS-IS and OSPF to make routing decisions based on the state of network links. Also called the <i>Dijkstra algorithm</i> .
SPI	Security Parameter Index. A portion of the IPSec Authentication Header that communicates which security protocols, such as authentication and encryption, are used for each packet in a VPN connection.
SPQ	Strict Priority Queuing. Dequeuing method that provides a special queue that is serviced until it is empty. The traffic sent to this queue tends to maintain a lower latency and more consistent latency numbers than traffic sent to other queues. <i>See also APQ.</i>
SSB	System and Switch Board. On an M20 router, Packet Forwarding Engine component that performs route lookups and component monitoring and monitors FPC operation.
SSH	Secure shell. Software that provides a secured method of logging in to a remote network system.
SSRAM	Synchronous Static Random Access Memory.
static LSP	<i>See static path.</i>
static path	In the context of traffic engineering, a static route that requires hop-by-hop manual configuration. No signaling is used to create or maintain the path. Also called a <i>static LSP</i> .
STM	Synchronous Transport Module. CCITT specification for SONET at 155.52 Mbps.

strict	In the context of traffic engineering, a route that must go directly to the next address in the path. (Definition from RFC 791, modified to fit LSPs.)
STS	Synchronous Transport Signal. Synchronous Transport Signal level 1. Basic building block signal of SONET, operating at 51.84 Mbps. Faster SONET rates are defined as STS- <i>n</i> , where <i>n</i> is a multiple of 51.84 Mbps. <i>See also</i> SONET.
stub area	In OSPF, an area through which, or into which, AS external advertisements are not flooded.
subnet mask	Number of bits of the network address used for host portion of a Class A, Class B, or Class C IP address.
summary link advertisement	OSPF link-statement advertisement flooded throughout the advertisement's associated areas by area border routers to describe the routes that they know about in other areas.
sysid	System identifier. Portion of the ISO nonclient peer. The sysid can be any 6 bytes that are unique throughout a domain.
System and Switch Board	<i>See</i> SSB.
T	
TACACS+	Terminal Access Controller Access Control System Plus. Authentication method for validating users who attempt to access the router using Telnet.
TCP	Transmission Control Protocol. Works in conjunction with Internet Protocol (IP) to send data over the Internet. Divides a message into packets and tracks the packets from point of origin to destination.
ToS	Type of service. The method of handling traffic using information extracted from the fields in the ToS byte to differentiate packet flows.
traffic engineering	Process of selecting the paths chosen by data traffic in order to balance the traffic load on the various links, routers, and switches in the network. (Definition from http://www.ietf.org/internet-drafts/draft-ietf-mpls-framework-04.txt .) <i>See also</i> MPLS.
transit area	In OSPF, an area used to pass traffic from one adjacent area to the backbone or to another area if the backbone is more than two hops away from an area.
transit router	In MPLS, any intermediate router in the LSP between the ingress router and the egress router.
transport mode	An IPSec mode of operation in which the data payload is encrypted, but the original IP header is left untouched. The IP addresses of the source or destination can be modified if the packet is intercepted. Because of its construction, transport mode can be used only when the communication end point and cryptographic end point are the same. VPN gateways that provide encryption and decryption services for protected hosts cannot use transport mode for protected VPN communications. <i>See also</i> tunnel mode.
Triple-DES	A 168-bit encryption algorithm that encrypts data blocks with three different keys in succession, thus achieving a higher level of encryption. Triple-DES is one of the strongest encryption algorithms available for use in VPNs.
tunnel	Private, secure path through an otherwise public network.

tunnel mode An IPSec mode of operation in which the entire IP packet, including the header, is encrypted and authenticated and a new VPN header is added, protecting the entire original packet. This mode can be used by both VPN clients and VPN gateways, and protects communications that come from or go to non-IPSec systems. *See also transport mode.*

Tunnel PIC A physical interface card that allows the router to perform the encapsulation and decapsulation of IP datagrams. The Tunnel PIC supports IP-IP, GRE, and PIM register encapsulation and decapsulation. When the Tunnel PIC is installed, the router can be a PIM rendezvous point (RP) or a PIM first-hop router for a source that is directly connected to the router.

type of service *See ToS.*

U

unicast Operation of sending network traffic from one network node to another individual network node.

UPS Uninterruptible power supply. Device that sits between a power supply and a router (or other piece of equipment) the prevents undesired power-source events, such as outages and surges, from affecting or damaging the device.

V

vapor corrosion inhibitor *See VCI.*

VCI Vapor corrosion inhibitor. Small cylinder packed with the router that prevents corrosion of the chassis and components during shipment.

VCI Virtual channel identifier. 16-bit field in the header of an ATM cell that indicates the particular virtual channel the cell takes through a virtual path. Also called a *logical interface*. *See also VPI.*

virtual circuit identifier *See VCI.*

virtual link In OSPF, a link created between two routers that are part of the backbone but are not physically contiguous.

virtual path identifier *See VPI.*

virtual private network *See VPN.*

Virtual Router Redundancy Protocol *See VRRP.*

VPI virtual path identifier. 8-bit field in the header of an ATM cell that indicates the virtual path the cell takes. *See also VCI.*

VPN virtual private network. A private data network that makes use of a public TCP/IP network, typically the Internet, while maintaining privacy with a tunneling protocol, encryption, and security procedures.

VRRP Virtual Router Redundancy Protocol. On Fast Ethernet and Gigabit Ethernet interfaces, allows you to configure virtual default routers.

W

wavelength-division multiplexing *See WDM.*

WDM Wavelength-division multiplexing. Technique for transmitting a mix of voice, data, and video over various wavelengths (colors) of light.

WFQ Weighted Fair Queuing.

weighted round-robin *See WRR.*

WRR Weighted round-robin. Scheme used to decide the queue from which the next packet should be transmitted.



Part 7

Index

- Index on page 359
- Index of Statements and Commands on page 363

.....

Index

Index

Bold numbers point to command summary pages.

Symbols

- #, in configuration statements..... xxiv
- ()
 - in syntax descriptions xxiv
- < >, in syntax descriptions..... xxiii
- []
 - in configuration statements..... xxiv
- { }, in configuration statements..... xxiv
- |
 - in syntax descriptions xxiv

A

- all (tracing flag)..... 25, 55
- application-based Layer 3 VPNs..... 172

B

- BOOTP service 26
- braces, in configuration statements..... xxiv
- brackets
 - angle, in syntax descriptions..... xxiii
 - square, in configuration statements xxiv

C

- carrier-of-carriers VPNs 261, 265
 - Internet service 267
 - LDP 320
 - VPN service 267
- CCC encapsulation
 - Layer 2 VPNs..... 18, 36, 48
- CE devices 3
- CE routers 3
- comments, in configuration statements xxiv
- community statement
 - origin option, usage guidelines 90
- connections (tracing flag) 25, 55, 333

- control-word statement
 - Layer 2 circuits 331
 - Layer 2 VPNs..... 47
- conventions, documentation xxiii
- curly braces, in configuration statements xxiv
- customer edge devices *See* CE devices
- customer edge routers *See* CE routers
- customer support
 - contacting xxvi

D

- DCU
 - Layer 3 VPNs..... 213
- description statement..... **47, 253**
 - Layer 2 VPNs
 - usage guidelines 17
 - Layer 3 VPNs
 - usage guidelines 86
- detail (tracing flag modifier)..... 55, 333
- documentation conventions xxiii

E

- EBGP, multihop for Layer 3 VPNs 105
- encapsulation statement..... **48**
 - usage guidelines 18, 19
- encapsulation types, Layer 2 VPNs 24, 50
- error (tracing flag)..... 25, 55, 333

F

- family inet-vpn option 85
- family l2vpn option 16
- FEC (tracing flag) 333
- filter-based forwarding, Layer 3 VPNs..... 172
- firewall filters, Layer 3 VPNs..... 172

l2vpn statement.....	51
usage guidelines.....	16
labeled-unicast statement	321
usage guidelines.....	269
Layer 2 circuits	
interface statement	331
usage guidelines.....	327
neighbor statement	332
usage guidelines.....	327
no-control-word statement	
usage guidelines.....	329
traceoptions statement	332
usage guidelines.....	329
virtual-circuit-id statement	334
usage guidelines.....	328
Layer 2 VPNs	
CCC encapsulation	18, 36, 48
configuration example	29
configuration statements.....	47–56, 331–334

Layer 2.5 VPNs.....	19
Layer 3 VPNs	
application-based	172
configuration statements.....	253–257
DCU	213
description statement	253
usage guidelines.....	86
example configurations.....	127
export policy	91, 256
filter-based forwarding.....	172
firewall filters	172
graceful restart	105
overview	4
GRE tunnels	198
IBGP session between PE routers	85
import policy.....	90, 256
inet6-vpn statement	253
usage guidelines.....	102

instance type.....86, 254
 interface between PE and CE.....86
 IPv6 configuration.....102
 LDP, enabling.....81
 loopback interface.....88
 multicast73, 105
 multihop EBGp and IBGP.....105
 OSPF domain ID97
 route distinguisher89, 255
 route-distinguisher-id statement255
 usage guidelines.....21
 routing instances.....85
 RSVP, enabling.....83
 SCU213
 signaling protocols, enabling.....81
 site of origin60
 target VPN.....60
 VPN of origin.....60
 vrf-table-label257
 vrf-target257
 LDP
 enabling for Layer 2 VPN signaling.....12
 enabling for Layer 3 VPN signaling.....81
 loopback interface, logical unit.....88

M

MPLS
 IGP shortcuts.....67
 multicast
 Layer 3 VPNs.....73, 105
 multihop EBGp and IBGP.....105

N

neighbor statement.....332
 usage guidelines.....327
 nlri (tracing flag)25, 55
 no-control-word statement
 Layer 2 circuits331
 Layer 2 VPNs47
 usage guidelines.....22, 329
 no-decrement-ttl statement
 VPNs26
 no-propagate-ttl statement
 VPNs26
 no-stamp option55, 333

O

OSPF domain ID.....97
 overlapping VPNs183

P

P routers.....3
 PE routers.....3
 IPSec110
 policer statement
 usage guidelines20
 policers
 Layer 2 VPNs.....20
 provider edge routers3
 provider routers.....3
 proxy statement.....52
 usage guidelines20

R

remote statement.....52
 usage guidelines20
 remote-site-id statement.....52
 usage guidelines24
 replace option.....55, 333
 resolve-vpn option.....321
 route (tracing flag)25, 55
 route distinguisher.....89
 Layer 2 VPNs.....21
 route limit, configure101
 route origin.....90
 route reflection, VPNs.....125
 route-distinguisher statement.....53, 255
 usage guidelines89
 route-distinguisher-id statement53, 255
 usage guidelines21
 routing instance
 configuring for Layer 3 VPNs.....85
 description statement47, 253
 usage guidelines17, 86
 Layer 2 VPNs.....16
 RSVP
 enabling for Layer 3 VPN signaling.....83
 Layer 2 VPNs.....14

S

SCU, Layer 3 VPNs.....213
 site configuration
 Layer 2 VPNs.....23
 site of origin60
 configuration90
 site statement.....54
 usage guidelines23
 site-identifier statement.....54
 usage guidelines23
 size option55, 333
 support, technical
 customer support, contactingxxvi

T	
target VPN.....	60
TCC encapsulation types, Layer 2 VPNs	19
technical support	
customer support, contacting.....	xxvi
topology (tracing flag).....	25, 55, 333
traceoptions statement	
Layer 2 circuits.....	332
usage guidelines.....	329
Layer 2 VPNs	55
usage guidelines.....	25
tracing flags.....	25, 55, 329, 332
all.....	55
connections.....	55, 333
error.....	55, 333
FEC	333
nlri	55
route	55
topology	55, 333
tracing protocol traffic	
Layer 2 VPNs	25
TTL decrementing	
VPNs	26
typefaces, documentation conventions	xxiii

V	
virtual-circuit-id statement	334
usage guidelines.....	328
VPN of origin.....	60
VPN tunnel interfaces.....	95
VPN tunnels	95
vpn-apply-export statement.....	256
usage guidelines.....	92
VPNs	
CE devices.....	3
IGP shortcuts.....	67
multiple instance LDP	320
overlapping.....	183
P routers	3
packet forwarding.....	26
PE routers	3
<i>See also</i> Layer 2 VPNs	
<i>See also</i> Layer 3 VPNs	
VRF table lookups	95
VRF tables	95
vrf-export statement.....	56, 256
usage guidelines.....	91
vrf-import statement	56, 256
usage guidelines.....	90
vrf-table-label statement	257
usage guidelines.....	94
VT interfaces.....	95
vrf-target statement.....	257
usage guidelines.....	93
VT interfaces	95

Index

Index of Statements and Commands

C control-word statement Layer 2 circuits 331 Layer 2 VPNs 47	S site statement 54 site-identifier statement 54
D description statement 47, 253	T traceoptions statement 55, 332
E encapsulation statement 48 encapsulation-type statement 50	V virtual-circuit-id statement 334 vpn-apply-export statement 256 vrf-export statement 56, 256 vrf-import statement 56, 256 vrf-table-label statement 257 vrf-target statement 257
I inet6-vpn statement 253 instance-type statement 51, 254 interface statement 51, 254, 331	
L l2vpn statement 51 labeled-unicast statement 321	
N neighbor statement 332 no-control-word statement Layer 2 circuits 331 Layer 2 VPNs 47	
P proxy statement 52	
R remote statement 52 remote-site-id statement 52 route-distinguisher statement 53, 255 route-distinguisher-id statement 53, 255	

